



量化投资专家系统 开发与策略实战

王昭东 著

电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书从软件工程的角度把一个看似不可能完成的大型金融软件系统抽丝剥茧、逐步完善,从而提高技术人员的大局观、合作意识和问题分解能力。全书一共5章,第1章介绍专家系统与量化投资专家系统,起到提纲挈领、点明任务主题的作用。第2章介绍量化投资专家系统开发,需求是设计更是算法,通用性极强。第3章从PHP开发者的角度详细讲解代表性模块的开发与实现,从而达到举一反三的目的,加深读者的印象。第4章从基本面、技术面和高频方面分别列举了两个策略。第5章通过一些小案例提高读者的开发能力,从而突破量化投资专家系统的单一局限。

本书适合金融从业人员、量化交易员、专家系统研究者、机器学习爱好者、金融科技研究者、大型Web系统架构师与设计师阅读。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

量化投资专家系统开发与策略实战 / 王昭东著. —北京: 电子工业出版社, 2018.4
(量化交易丛书)

ISBN 978-7-121-33315-6

I. ①量… II. ①王… III. ①金融—专家系统—系统开发—研究 IV. ①TP182

中国版本图书馆CIP数据核字(2017)第311530号

策划编辑: 黄爱萍

责任编辑: 牛 勇

印 刷: 三河市双峰印刷装订有限公司

装 订: 三河市双峰印刷装订有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路173信箱 邮编: 100036

开 本: 720×1000 1/16 印张: 18 字数: 267千字

版 次: 2018年4月第1版

印 次: 2018年4月第1次印刷

定 价: 69.00元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: (010) 51260888-819, faq@phei.com.cn。

推 荐 序

金融的本质就是风险控制，而风控的好坏直接影响金融公司的长远发展战略，就传统技术而言，风控主要靠一两个顶级专家的经验，但是这些专家少之又少，工作效率也有待提高，那么怎么办？人工智能出来了，我们能不能把专家的思想复制到很多台计算机上？如果能，我们又应该怎样复制？复制的结果是不是理想？这就需要我们非常务实地做一连串的调研、分析、设计、开发和评估工作。就目前来看，我们迫切需要一本具有指导意义的技术图书，供从业人员翻阅，而本书就是这样一本非常值得业内人士阅读的图书。

量化投资专家系统是人工智能在科技金融领域的一个具体应用。我认为，人工智能作为当前最先进的技术工具之一，其商业化最有价值的行业就是金融行业，而其中最好的场景则属智能风控。智能风控关系到金融公司的方方面面，小到买卖股票，大到投资并购，无一不是。金融公司甚至会借助人工智能的技术优势，创建全新的金融巨无霸。

本书所写的量化投资专家系统已经超越了传统金融专家系统，量化投资专家系统融合了办公自动化、客户关系管理、机器学习、增强学习、遗传算法和脑电测试等技术，使静态的金融专家系统变成可以自我升华的量化投资“大家”。

最后，希望读者能够借助科技的力量进行更深入、更全面、更科学、更合理的资源整合。

熊猫金控创始人 赵伟平

前言

人工智能作为一个空前强大的力量，正在全面改变着传统的金融领域，传统的金融要么选择改变，要么高贵地逝去。就拿股票交易这个行业来说，机器人与人相比就有着速度、成本、工时和理性的优势，如图 1 所示。

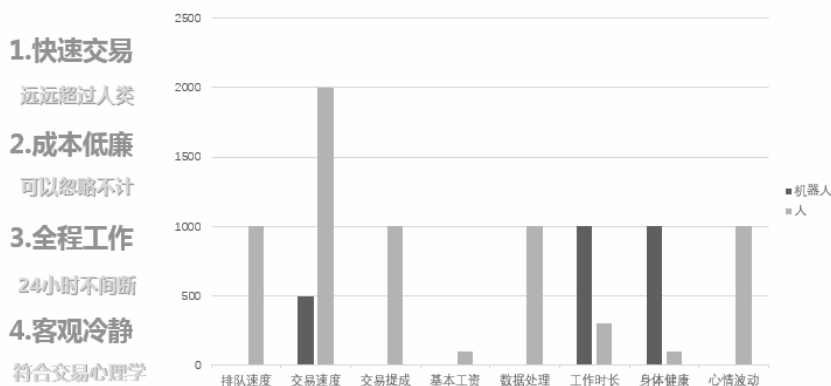


图 1 机器人与人的交易优势对比

图 1 所展示的仅仅是机器人与人的交易优势对比，试想一下，如果机器人能够利用机器学习（尤其是深度学习）技术进行策略生成，那么传统的人类分析师和交易员又该如何？所谓的明星交易员还会存在吗？全自动股票交易过程如图 2 所示。



图 2 全自动股票交易过程

那么，我们如何才能做出如图 2 所示的那样全自动的量化投资专家系统呢？本书将讲解如何利用人工智能中的专家系统技术开发一个全自动的量化投资专家系统。

轻松注册成为博文视点社区用户（www.broadview.com.cn），扫码直达本书页面。

- **提交勘误：**您对书中内容的修改意见可在 [提交勘误](#) 处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- **交流互动：**在页面下方 [读者评论](#) 处留下您的疑问或观点，与其他读者一同学习交流。

页面入口：<http://www.broadview.com.cn/33315>



目 录

第 1 章 专家系统与量化投资专家系统	1
1.1 专家系统	1
1.2 量化投资专家系统	11
第 2 章 量化投资专家系统开发	13
2.1 任务申请	13
2.2 可行性研究	15
2.3 需求说明	21
2.4 项目开发计划	25
2.5 产品原型图	28
2.6 模块开发说明	52
2.7 数据库设计	69
2.8 网页设计	90
第 3 章 软件开发	98
3.1 基本框架	99
3.2 登录模块	107
3.3 会员管理	111
3.4 动态权限管理	115
3.5 自定义菜单与快捷菜单	123
3.6 站内信与动态流程	133
3.7 任务模块	137

3.8	系统维护	146
3.9	任务轮询	150
3.10	策略管理	154
3.11	数据统计	163
3.11.1	数据接口	163
3.11.2	数据查询	164
3.11.3	统计视图	170
3.12	网络爬虫	176
3.13	图片管理	181
3.14	机器学习	193
3.15	验收报告	195
第 4 章	经典策略解析	197
4.1	基本面分析	207
4.2	K 线	212
4.2.1	捉腰带	213
4.2.2	回马枪	214
4.2.3	锤子线	214
4.3	多因子	215
4.3.1	随机策略	215
4.3.2	多因子模型构建思路	217
4.4	小海龟	218
4.5	高频交易	222
第 5 章	项目拓展	224
5.1	模拟交易系统	224
5.2	海量图片生成系统	245
5.3	机器学习系统	262
附录 A	实际问题	274
附录 B	汉语言	275
附录 C	源码	278

1

第 1 章

专家系统与量化投资专家系统

1.1 专家系统

很久以前几乎所有的智能都是伴随着生命而存在的，这种传统的智能我们称其为生物智能，后来计算机出现后，人们才开始尝试用电子计算机等人工的方式创造智能，可以说现在的人工智能是与计算机的发展密不可分的，其中计算机之父图灵先生就在第二次世界大战以后出版了《智能型机器》和《电子计算机与智能》两本著作，书中着重讨论了如何用电子计算机来实现智能的问题，从此人工智能的序幕也被正式拉开了，发展到目前如火如荼的强人工智能时代。

早期人们对人工智能并不像我们现在对强人工智能或超级智能这么奢望，人们主要希望人工智能能进行一些诸如问题的求解和模式识别这样简单的工作，后来才开始了自然语言处理、博弈和自动程序设计。

在研究的过程中科学家普遍发现：人们之所以能够成为某个领域的专家，关键就在于他掌握了关于该领域的大量知识，这些知识一部分是从书本上或他人那里学来的，另外一部分是在长期的实践中逐渐积累起来的，正是这些经验才让专家比普通人技高一筹。

那么，我们能不能利用计算机技术把专家的这种优势发挥出来呢？答案是肯定的。专家系统就是这样一种利用计算机存储某个领域的大量知识，并能够有效地利用这些知识去解决问题的系统。

因此，科学家参照人类的逻辑思考过程，创造了以心理过程为主导的逻辑主义（也叫符号主义），逻辑主义特别强调逻辑和推理。而推理则取决于知识。一般来说，人工智能根据解决问题的不同，所需要的知识也是不同的，一般的人工智能解决一般的问题需要一般的知识，专业的人工智能解决专业的问题需要专业的知识。而这种专业的人工智能就是专家系统（另一种是通用专家系统），也正是因为这个原因专家系统才变得与专家密不可分，比如工程设计专家系统、化学专家系统、地质勘探专家系统、农业病虫害专家系统和内科疾病诊断专家系统等。

最早的专家系统主要是使用 LISP 或 Prolog 语言设计的，但是实际上任何一个专家系统发展到后来都是一门理论性和实用性很强的交叉学科，它有时候并不追求所谓的“逻辑与智能”，而是更多地强调人机交互。因为专家系统的使用者往往也有很强的专业知识，所以现在的专家系统做的是批处理、大计算的辅助决策工作。

专家系统是人工智能的一个重要分支，目前专家系统经过长达 60 年的发展已然成为一门非常成熟的技术，广泛地应用于数学、天气、金融、农业、地理、医疗、交通、军事和化学等领域。专家系统的目的就是模拟专家来完成日常工作的全过程，从而实现人人都是专家的梦想。

在专家系统中，专家的知识起到了非常重要的作用，在知识就是力量的今天，**专家系统的价值已然等于知识的数量乘以质量**，也就是说知识的数量越多，其质量就越高、价值就越大。

专家的知识虽然宝贵，但是缺乏严谨的理论依据，很难保证在各种情况下都适用，所以专家知识只是在一定条件下有效，换句话说就是专业知识适用于专业领域，而不具备科学性。所以，推崇科学教育的近代学校，一般是不会将专家知识作为教科书内容进行讲解的，尽管在实际工作中非常有效，也是如此。

专家知识更多的是一种经验，而经验又都是不确定的，它的发生有一定的概率分布。比如金融（股票投资）专家系统，就是靠概率来获得收益的。

专家系统多种多样，一般都是按照应用领域进行分类的，比如量化投资专家系统就是金融类的，而如果按照用途分类的话，专家系统还可以分为解释型、预测型、诊断型、调试型、规划型、设计型、监督型、控制型和教育型等，其关系图如图 1-1 所示。

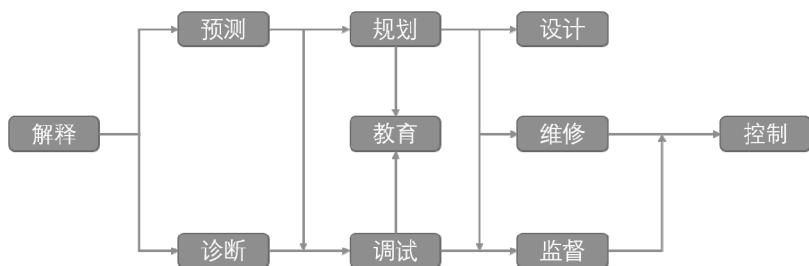


图 1-1 专家系统用途关系图

一般气象、军事、人口、交通、财政、经济等属于预测型；图像分析、集成电路分析、染色体分析等属于解释型；医疗诊断、电子仪器失效诊断等属于解释型；辅助教学、聋哑人语言训练等属于教育型。

而我们刚才所说的量化投资专家系统一般都是预测型（尤其是和投资相关的部分）的。一般来讲，预测型的专家系统主要是以似然推理或不精确推理为主的，因为我们所依据的数据会随着时间的变化而变化，而我们的目的则是“能够描述和模拟各种数据随时间变化而变化的情况、各个事件在时间关系上的先后顺序及其所在时间内关联事件时所产生的影响”。

所以，这就要求量化投资专家系统既能够处理不完整的信息，又能够在不确定的未来做出正确的选择。为了更好地实现上述目的，我们通常的做法是一方面利用各种渠道获得有效数据，另一方面注重长远预测与近期预测的相互影响。

虽然每位专家的专注领域不同，但是专家之所以能够成为专家，还是有很多共性的，我们通过计算机技术可以把这种共性结构化，比如股票投资专家系统就是一个把交易员的思想转换成计算机可以理解的代码并执行

的过程。

我们通过计算机技术完全可以把交易员常用的数据分解成知识库、行情库、支持库和交易指令库四个部分。

我们通过计算机依据**知识库**并借助**支持库**对实时的**行情库**进行推理，最后根据推理的结果决定是否调用具体的**交易指令**。

专家系统一般包括三个方面，分别是知识库、推理机和解释器。

(1) 知识库

知识库是指那些经过验证的真实知识或经验，哪怕你的知识是一个假设，这个假设也必须是真实的，否则不能记录在知识库体系中。比如假设“ $1+1=2$ ”或“ $1+1=10$ ”，那么这个假设必须是真实的，所谓的真实就是指在一定条件下要么为真、要么为假，而不能同时既为真又为假，否则即使有一丁点的歧义，后续都将无法进行推理和演绎。

也正是这个原因，知识库里面的知识特别强调一切要从事实出发，而且知识库中至少有一个事实。比如“**它是动物**”，那么这个知识就要百分之百正确，不能有任何不确定性。

知识库对专家系统的重要性不言而喻，不过知识库干得更多的是体力活的工作，在实际工作中更多的是支持工程师的工作，对于程序员而言，我们接触的很少。

(2) 推理机

推理机是什么呢？推理机实际上就是我们对一些事情有自己的理解和判断，是一个判断过程。比如，如果这个动物长羽毛，那么我们判断它是鸟类。我们是怎么知道它是动物而不是植物的呢？因为前面的知识库已经百分之百地告诉了我们它是动物，所以我们才有后面的推理。

其实很多时候，推理机和知识库是相辅相成的，知识库为推理机提供事实依据，推理机推理出来的结论又可以成为新的事实存入知识库。

至于推理的过程，我们可以采用正向推理，也可采用逆向推理，甚至双向推理。一般从事实出发，沿着规则最终得出结论的方法就是正向推理，

而从目标出发，沿着规则最终找到事实的方法就是逆向推理。另外，从统计中找出规律，再把规律推广后得出结果叫作归纳推理。不管哪种推理，量化投资专家系统都强调其结果的最优解。

在众多的解中找到一个最优解，一般都是多因素构成的，需要考虑这些指标对解的综合影响。一般来讲，最优解可以交给决策者自己来判断，受到最终决策者的主观影响，这个也是专家系统的一个非常重要的地方，就是辅助决策。另外，比较常用的是综合指标法，综合指标法就是把多因素决策换成单因素决策，然后变成雷达图，尤其是在多个专家决策时，可以设置不同的权重进行选择，如图 1-2 所示。

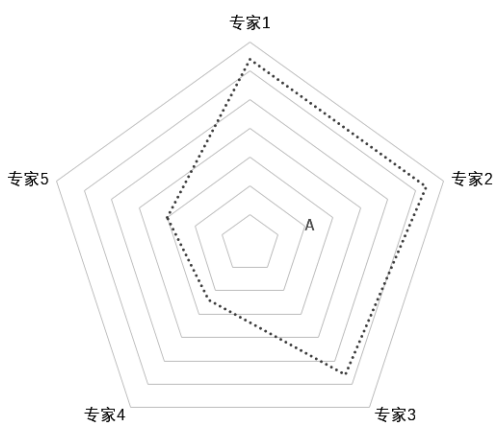


图 1-2 雷达（综合指标）决策图

虽然还有一些非常有趣的推理机，比如那些颠倒语句、偷换概念的诱导性回答，在量化投资专家系统里一般是不用的，或许这些在心理治疗专家系统中才可能看到。

患者：我很难过。

机器人：你为什么难过？

患者：突然感觉很想念他。

机器人：他是谁？

患者：我的一个朋友。

机器人：那么我们说说你的朋友吧。

.....

(3) 解释器

解释器同样是专家系统不可或缺的一部分。这个系统是给人用的，既然要和人打交道，就要让人们知道怎么用，还要告诉人们是怎样做出这样的结论的。

那么，这种告诉人们系统是怎样推理的过程或与使用者之间的交互过程就是解释器。

正是解释器的这个优点，才让它经久不衰，既让人们知道结果，又让人们知道原因，即**知其然并知其所以然**。

就拿刚才的例子来说，用户可能会问：“你怎么知道这个动物是鸟类呢？”答：“因为它身上长羽毛，所以它是鸟类。”

但是解释器并不像程序员理解的这么简单，因为很多专家和用户并不都是编程高手，他们可以用自己行业的语言描述清楚，并不表示他就能用计算机语言编写完整。为了解决这个问题，科学家提出了一种面向自然语言的编程语言（也叫第五代编程语言），比如 LISP 和 Prolog，不过真正意义上的第五代编程语言至今为止尚未出现。一般来讲，第一代编程语言是机器语言，面向机器设备；第二代编程语言是汇编语言，面向指令（如 Win32 汇编）；第三代编程语言是高级语言，面向过程（如 C、）或面向对象（如 C++）；第四代编程语言是框架语言，面向应用（如 .NET——以 C# 为主、面向 Windows 应用程序，TensorFlow——以 Python 为主、面向人工智能，Laravel——以 PHP 为主、面向 Web 开发，Hibernate——以 Java 为主、面向企业应用）；第五代编程语言是智能语言，面向自然语言。

自然语言（第五代编程语言）目前制约着专家系统的全面普及并影响人工智能的发展，为了解决这个问题，量化投资专家系统可以尝试从以下几个方面进行突破。

(1) 代码高亮和语法提示。

- (2) 可以使用母语进行一定意义上的编程。
- (3) 自动语法纠正和源码格式化。
- (4) 可以人为修改已有函数的映射关系。
- (5) 拥有与汉语言相似的语法结构。
- (6) 框架尽量做到可视化编程。
- (7) 弱类型语言，对变量、大小写甚至语法结构不敏感。
- (8) 将金融常用的模型算法组件化。
- (9) 有完备的教程和示例代码。
- (10) 拥有一个相对完善的社区。

也就是说，我们尽量让计算机自己去完成编程的工作，用户只要表达清楚自己的意图就可以了。至于 1 和 1.00 是什么变量类型那不是用户该考虑的问题。

其实，解释器还有一个努力发展的方向就是要求这个语言有学习的功能，不过就目前来看，还有很长一段路要走。

综上所述，专家系统的三大要素如图 1-3 所示。

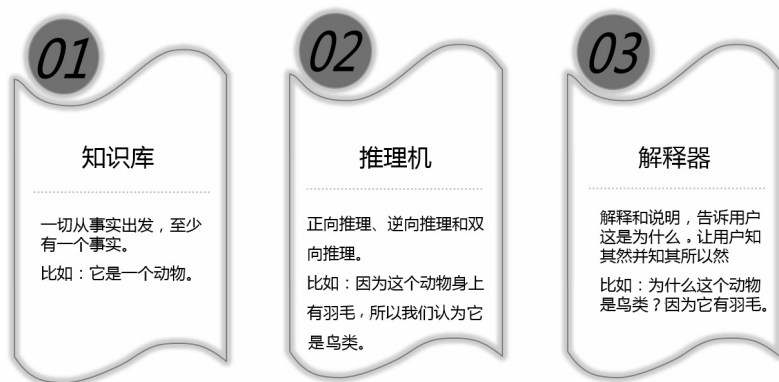


图 1-3 专家系统的三大要素

前面讲解了专家系统的三大要素，那么真正的专家系统大概是怎样的流程和结构呢？专家系统的三大要素的流程图如图 1-4 所示。

专家在专家系统中仍然扮演专家的角色，只不过这个专家需要通过人机交互界面让系统理解专家的知识和思想。比如，这是一个量化投资专家系统，那么金融专家就需要通过人机交互界面把一些公式和模型录入到知识库，或者告诉专家系统从哪里导入并抓取值得信赖的知识，这个过程对于金融专家而言可能有些庞杂，所以我们可以找那些既懂计算机编程又懂一些金融知识的工程师来做这件事。

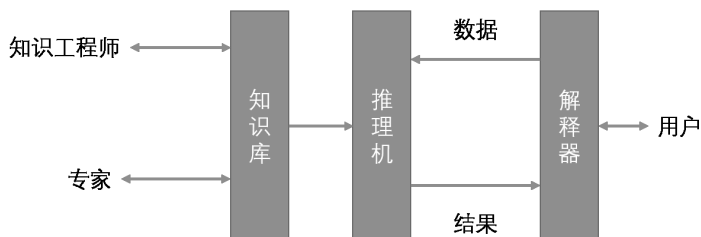


图 1-4 专家系统的三大要素的流程图

当金融专家录入知识后，金融用户就可以使用了，金融用户通过人机交互界面与系统的数据库和推理机打交道，所以量化投资专家系统真实的流程图可能如图 1-5 所示。

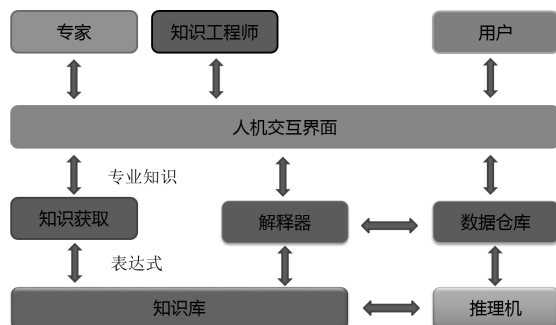


图 1-5 量化投资专家系统真实的流程图

我们这里举一个例子。比如，一个量化投资专家系统有三个专家，并且三个专家都是炒股高手，但是他们具体的思想有所不同。

专家 A 认为：当均线出现金叉时买入。

专家 B 认为：当布林线向上突破 80 时买入。

专家 C 认为：当成交量大于 100 手时买入。

用户在使用专家系统时发现均线出现了金叉，并且布林线向上突破了 80，而且成交量大于 100 手，于是他就兴高采烈地买入了。既然三个专家都说要买入，那就听专家的好了。

但是你会发现，事实上随着专家人数的增加，专家的意见可能相反，或者很难同时满足所有专家提出的条件，怎么办？这时就可以通过给每个专家打分来设置他的权重。

比如，你认为专家 A 很重要，给他打 60 分；专家 B 也很好，给他打 30 分；专家 C 感觉一般，给他打 10 分。

这样量化投资专家系统就把海量的用户和众多的专家结合起来了。如果我们把金融专家换成数学专家或化学专家，那么这个专家系统就是数学专家系统或化学专家系统了。

专家系统与传统计算机程序相比优势是很明显的，下面看一下两者的对比，如表 1-1 所示。

表 1-1 专家系统与传统计算机程序对比

列项	传统计算机程序	专家系统
处理对象	数字	符号
处理方法	算法	启发式
处理方式	批处理	交互式
系统结构	数据和控制集成	知识和控制分离
系统修改	难	易
信息类型	确定性	不确定性
处理结果	最优解	可接受解
适用范围	无限制	封闭世界假设

讲了这么多专家系统的知识，最后给出一个专家系统发展的大致时间表，供读者参考。

1937 年，图灵发表了理想计算机的论文，提出了著名的图灵机原型。

1954 年，提出控制规则执行的马尔可夫算法。

1956 年，达特茅茨会议，这一年也被称为 AI 元年。

1957 年，提出通用问题求解办法。

1960 年，麦卡锡发明了 LISP 编程语言。

1965 年，斯坦福大学的 Feigenbaum 和 J.Lederberg 开发出 DENDRAL（化学专家系统），这是世界上第一个专家系统。

1969 年，开发出 MACSYMA（数学符号运算专家系统）。

1972 年，开发出 MYCIN（传染病诊断系统）和矿床探测专家系统。

1974 年，出现 OPS（人工智能编程语言）。

1975 年，明斯基提出了框架知识表示法。

1977 年，Feigenbaum 等人提出了知识工程的概念，知识工程是人工智能的原理和方法。出现了 EMYCIN（医生专家系统）、CLIPS（C 语言集成产生式系统）、G2（实时智能专家系统）。

1978 年，开发出 R1（通用专家系统）。

1979 年，开发出 Rete 算法（模式结构匹配）。

1982 年，日本宣布开发第五代机器人（以 Prolog 为开发语言）。

1983 年，开发出 KEE（综合多样知识表现与推理方法的专家系统）。

1985 年，NASA 开发出 CLIPS 专家系统工具，我国开发了砂姜黑土小麦施肥专家咨询系统（我国第一个农业专家系统）。

1988 年，开发出以神经网络为基础的专家系统。

1989 年，日本宣布实施人类领域科学计划（第六代计算机）。

1990 年，大量专家系统不断涌现，但维护成本居高不下。

2000 年以后，专家系统协同办公软件开始全面普及。

1.2 量化投资专家系统

量化投资专家系统是专家系统的一个分支，也有人认为，量化投资专家系统与数学专家系统一样非常容易获得成功，从两个系统都和数字打交道来看，量化投资专家系统的设计和开发并不是特别困难，但是在具体应用效果上还是存在很大差异的，最终的结果就是量化投资专家系统在大型金融机构中表现良好，而在小企业中表现差强人意。其原因一方面来自于专家的知识库，另一方面来自于专家系统的完整性，因为小企业很难开发和维护一套完整的量化投资专家系统。

从原理上来说，量化投资专家系统并不复杂，以股票投资为例，有的专家认为股票市盈率浮动于行业市盈率 $\pm 30\%$ 是正常的，所以专家建议低于行业平均市盈率 10%时建仓 10%、低于行业平均市盈率 20%时建仓 30%、低于行业平均市盈率 30%时建仓 10%，如表 1-2 所示。

表 1-2 根据市盈率建仓

低于行业平均市盈率	10%	20%	30%	>30%
仓位选择	10%的仓位	30%的仓位	10%的仓位	持仓观望

上述专家其实就做了一件事，这件事就是有限单元法。有限单元法就是把数据分割成我们需要的元数据，这种元数据是一个有限小（记住不是无限小）的数据，就像物质是由原子和电子构成的一样（即便还可以继续往下分，但是不需要继续分了，因为原子和电子已经足够用了），我们需要把各种杂乱无章的数据分割成恰到好处的元数据，然后由这些元数据构成复杂的数据集。

比如，上市公司财报中每笔合同的金额、单位和类型等。

当然，要把这种庞大的数据分割成一个个有意义的数据是一个非常复杂的过程，不仅仅是 NLP（自然语言处理）这么简单，因为还有很多报表和真假难辨的票据合同。有时我们把这种看似连续的数据进行有限的分割是可以通过我们已有的经验来划分的。

比如股票涨跌的幅度，我们分为 1%、2%、3%、4%、5%、6%、7%、8%、9% 及 9% 以上 10 个档位。而这种看似主观的分法又可以分为普通有限元数据（有限条、有限层）、随机有限元数据、模糊有限元数据和模糊随机有限元数据，具体怎么分就看专家的经验了。

另外，在量化投资专家系统中，很多金融模型实际上都是已知的，在真正工作中，我们更多的是让系统进行参数估计和识别，从而达到一定的优化目的。

比如，我们买入一手股票作为输入，股价对应的反应是输出，从而建立一个动态模型，并确定其中的参数，如价格、数量、时间、涨幅、涟漪效果和阻尼系数等。

也就是说，通过已知模型的实测反应来确定某只股票某个时间点的资金承载量。这样做的好处是尽量避免造成股价较大的波动，从而影响股价。

具体方法可以使用表格、代数方程、积分方程等进行表示，这些方法通常还可分为参数模型与非参数模型、线性模型与非线性模型、确定性模型与随机模型、连续模型与离散模型、时变模型与非时变模型等。

至于最终做出的参数估计是否符合实际需求，我们常用的判别标准就是最小二乘法、极大似然法和极大熵法等。

我们知道金融投资都有一定的不确定性，这种不确定性主要是因为缺乏一个可明确定义和评定的标准，因此边界很难界定清楚。

为了减小这种模糊性和不确定性，我们建议把这些数据进行量化，可以说量化是目前量化投资专家系统的一个非常重要的方法，另外一个重要的方法就是机器学习。

本书面向对金融专业有一定了解的开发人员，所以读者最好拥有一定的股票、期货投资交易经验，否则可能会影响进一步阅读和使用。

读者最起码要知道证券交易所、开户银行、交易软件是什么，以及明白股票和期货挣钱的原理。

2

第 2 章

量化投资专家系统开发

2.1 任务申请

量化投资专家系统是一项大型的软件工程，为了保险起见，大型工程一般采用瀑布开发模式进行开发，瀑布模式非常注重项目的可行性和总体规划，也正是因为这个原因，瀑布模式才可以严格控制具体软件工程的生

产开发环节，瀑布模式如图 2-1 所示。

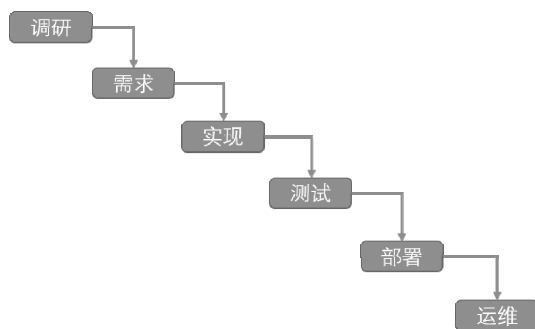


图 2-1 瀑布模式

我们要做一个量化投资专家系统，单凭自己的力量是无法完成的，这时可能需要更多人的帮助，尤其需要争取到主管领导的同意，那么就需要

填写一个任务申请表，就像要采购服务器一样，说明为什么要这么做，以及这么做的好处。任务虽然不用说得很细，但是要把大概脉络说清楚，如果有标的产品那就更好了。

阅读范文是非常有必要的，尤其是作为一个项目的管理者和核心成员必须要了解你做的产品否则就是闭门造车。本章所有的范文都是根据项目实际情况编写的，有较强的参考价值。

任务申请的范文如下。

任务申请

经过半年多的调查和研究，我认为量化投资专家系统是比较可行的，尤其是一个智能、灵活、高频、稳定的程序化交易系统。

目前，高盛、桥水和城堡基金等公司都已经拥有了自己的量化投资专家系统。我公司现在已经有几套不错的交易策略用于股票和期货市场，出于对策略的扩展性和保密性的要求，我强烈建议打造一个属于我们自己的量化投资专家系统。

我相信，系统在不断完善后，可以为我公司源源不断地创造财富，而未来其应用范围也不仅局限于金融行业。

由于该项目处于萌芽阶段，所以建议使用自有资金，这样可以最大化保护项目的能动性，系统以后的发展视自身的利润自行决定。

申请人：王昭东

申请日期：2017年9月8日

审核人：总经理

批准日期：2017年9月8日

当然，很多时候，这样大型的项目很可能是领导授意的，或者是领导主持的多次讨论会最终确定的结果，读者只是这个项目的具体负责人而已，但是不管是主动还是被动接受这个项目，任务申请这个过程最好还是要有，任务申请被审批后，就表示一个项目正式开始了。

2.2 可行性研究

正常情况下，申请一个项目是需要论证其可行性的，或者这个可行性的报告在若干次会议中已经成型，其主要目的是“说服”相关主管领导同意这个项目的实施，另一方面也是对自己信心的一个加强。所以，可行性报告尽量做到真实可信，语句通顺。

可行性研究报告范文如下。

量化投资专家系统 可行性研究报告

拟制人：王昭东

审核人：副总经理

批准人：总经理

2017年9月30日

一、引言

（一）编写目的

量化投资专家系统是一个复杂而庞大的工程，像这样一个工程在创建之初就应该有所计划，否则很容易偏离目标。

编写的目的主要有以下5点。

（1）**有的放矢**，正因为有了该系统的可行性研究报告，才能把以前杂乱的想法进行汇总，将无序的资料进行整理，从而让自己更加清楚在做什么和要做什么。

（2）**与人交流**，有了这样一个可行性研究报告，在与人交流的时候能够方便很多，这样不仅减少了交流的时间，而且提高了交流的效果。

（3）**自我修正**，量化投资专家系统不是一朝一夕就能够开发完成的，

开发这样一个系统可能需要数年的时间，而作为这个系统的发起人，我们只能掌握开始，却不一定能够控制结局，为了系统本身的发展着想，应该与时俱进，有改变计划的心理准备。

(4) **指导思想**，团队总有人进进出出，当主创人员由于某种原因离开后，希望留下一些资料与后继者进行很好的沟通，指导其发展壮大。

(5) **不时之需**，整理这些资料以备不时之需。

(二) 背景

(1) 软件名称

量化投资专家系统。

(2) 相关人员

- 申请人：xxx
- 设计师：xxx
- 开发者：xxx
- 测试员：xxx
- 所有人：xxx

(3) 相关技术

硬件：计算机、GPU、互联网

软件：Windows、Linux

源码：PHP、HTML、JavaScript

数据：MySQL、MongoDB

(三) 定义

由于面向的都是专业人士，所以这里的定义主要是为了防止产生歧义，并没有特别的要求。

(四) 参考资料

略。

二、可行性研究的前提（背景）

目前，人工智能是一个与数字打交道的科学，而专家系统作为一个更加科学严谨的人工智能分支，尤其善于处理大数据和科学计算。人工智能是数字，金融也是数字，所以两者结合起来更加容易，项目落地也更加直接。

另外，我们做研究、开公司都需要挣钱，量化投资专家系统的主要目的就是挣钱，事实证明量化投资专家系统也确实大型金融财团盈利的不二法门。

目前，已经有过一些策略研究，并且模拟盘、实盘效果都不错，尤其是高频交易部分有值得一试的价值。

（一）要求

项目最好采用 B/S 结构，这样全国各地的用户使用起来才会更加方便，不用总是更新升级、打补丁。另外，这是一个企业级项目，需要独立的服务器集群，项目只适合集团内部使用，不能对外，也就是说这是一个企业私有云项目。

（二）目标

- （1）能够取代现在杂乱的协同管理软件。
- （2）能够辅助交易员进行交易决策。
- （3）形成比较完善的金融专家知识库。

（三）条件、假定和限制

暂时不能进行深度学习，有些长工期的项目需要分期设计开发。

（四）进行可行性研究的方法

项目主要根据我们自身的实际情况，经过多次的会议讨论和调查研究，最后提出综合解决方案。

（五）评价尺度

用户主观打分和交易员绩效客观对比，另外还要看成本的减少尺度。

三、对现有系统的分析

（一）美国

在美国有报道称，距离 40% 的金融领域员工被取代可能只剩下 8 年的时间，而作为主角的量化投资则成为人工智能的主要舞台。全球最大的资产管理公司贝莱德在 2017 年 3 月宣布裁撤包括 7 名投资经理在内的 100 名主动型基金部门员工，在涉及变动的 300 亿美元的资产中，有近 60 亿美元将由量化基金接管。据 Opimas 的分析，贝莱德采用人工智能取代传统人工的战略将给公司的总体运营成本带来 28% 的下降空间。

在投资上，桥水公司早就实现了机器交易，如桥水公司的最高端基金产品——“绝对阿尔法（Pure Alpha）”就使用投资算法购买和出售股票、债券、货币和其他资产。

科技创新也许正是桥水基金 40 多年屹立不倒的法宝——桥水基金由达里奥创建于 1975 年。

（二）人工智能的发展，将让许多投资银行节省大笔成本

对许多公司来说，这可是一个天大的好消息。根据联盟统计，在销售、交易和研究领域，全球 12 个最大的投资银行员工的平均年薪为 50 万美元，包括薪资和奖金。而联盟研究主管 Amrit Shahani 说，75% 的华尔街薪酬给了这些高薪“前端”员工。一旦这些交易员被机器取代，他们以前获得的薪酬将直接计入公司利润。毫无疑问，如果能够替公司节省一大笔花费，那么许多投资银行还是愿意做的。

事实上，自动化正在高歌猛进。在一些交易领域，如货币、期货和股票，人们已经不在所在地的交易所进行交易了。

（三）人类在逐渐退出一些工作领域，人工智能将发挥极大的作用

高盛已经开始推行自动化货币交易，并发现 4 个交易员可被 1 个计算机工程师替代。

Amrit Shahani 在哈佛的大会上说：“现在，高盛员工总数的三分之一是计算机工程师。”他还表示，投资银行的任务接下来会被自动化取代，包括传统上一直专注于人力技能和建立关系的工作。虽然这些人不会被完全取代，但是流程的自动化已经在逐步实施。现在，高盛的新消费贷款平台旨在整合信用卡余额，完全由软件运行，没有人为干预。

另外，彭博社报道称，摩根大通公司开发了一款金融合同解析软件 COIN。这款软件将 360 000 小时才能完成的工作压缩到几秒钟完成，而且错误率大大降低，它还不用休假。COIN 仅仅是摩根大通公司开发的 2000 个技术项目之一，科技对摩根大通公司的重要性不言而喻。

在过去两年里，机器人自动化崭露头角，可以预测某些结果并为客户提供新的解决方案，从而排除人为错误。正是由于这项技术获得了客户的认可，所以下一阶段摩根大通打算让机器学习和人工智能替代人工操作交易，从而进一步消除人工错误带来的风险。

不仅基金公司如此，明星交易员也在充分利用这项技术，比如保罗·都铎·琼斯早就开始大量投资了人工智能驱动的“量化”对冲基金了。

（四）缺点

长期以来，对冲基金都依赖计算机来帮助交易，这通常需要数据科学家使用计算机来构建大型的统计模型。这些模型非常复杂，从某种意义上说它们是较静态的。当市场发生变化时，它们的效果或许不会再像过去那样好。

因此，各个基金公司的方法就是一方面让人工智能成为投资顾问，另一方面利用机器学习尤其是深度学习，让人工智能不断进化。

（五）我国现状

我国的一些基金公司正在开始类似的尝试，但无论是资金投入、业务占比，还是资管规模，都与美国甚至欧洲的基金公司相差甚远。

虽然人工智能神通广大、无所不能，但它们背后的构思、设计、开发、更新、升级、检修、维护等，每一个环节都离不开人，是人创造了人工智能。近年来，不少明星交易员也纷纷开始向机器学习，希望能够

与人工智能共舞。这种双向互动的结果将是双赢的。

至于人工智能是否能够取代明星交易员，这个问题其实并不重要。因为在实盘交易中，人工智能与明星操盘手并非像围棋棋手那样对阵博弈，他们之间也并非“零和游戏”，其共同的对手都是大盘。而随着市场的不断发展，主观交易、程序化交易、人工智能交易将日趋交叉覆盖。

（六）人才

自从2017年3月邓力加入城堡基金并担任首席人工智能学家后，乔丹也正式受聘成为蚂蚁金服科学智囊团的主席，目前国内外的金融机构正在世界范围内向这些人工智能大咖们伸出橄榄枝。

四、所建议的系统

我们建议采用B/S架构的方式开发集团内部的量化投资专家系统，一方面用于整合内部资源，另一方面用于辅助投资决策，并尝试机器学习和自动化交易。与人类相比，计算机适合在短期和中期做出交易决策，不过由于机器学习需要大量的结构化数据，所以我们需要对接专业的金融数据接口。同时，我们可以通过强化学习和对抗学习的方法优化已有的策略和权重。最后建议尽量招聘那些有开发经验的人，从而保证整个项目的开发周期。

五、可选择的其他系统方案

无。

六、投资和效益分析

投资和效益分析需要我们详细计算和估算，每个项目都有自身的计算方式，不论是细水长流还是厚积薄发，大体上都离不开支出、收益和维护成本三个方面。其中支出包含设备成本、数据成本和人员成本等。

七、社会因素方面的可能性

国内A股市场暂时还没有完全开通量化交易，不过有些券商却提供了主经纪商业务，这个业务还是可以谈的，如果是量化交易，我们建议做期货、港股和外汇。

八、结论

我们的结论是：学习国外的先进技术，利用本土化的优势，完善自有的量化投资专家系统，从而保护集团的投资利益，同时进一步完善金融职能，促进市场稳定。

2.3 需求说明

在《可行性研究报告》出来之后，企业的最终决策者一般就会决定做这个项目了，但是这个项目到底要怎么做？需要结合公司的实际情况设计一个《需求说明书》，《需求说明书》设计的方法有很多，有人喜欢套用模板，根据《可行性研究报告》中的内容把其中的相关名词替换掉，从而快速完成一个《需求说明书》，但是这种方法过于死板；还有一种方法需要通过多次会议讨论，由于项目比较大，且可以分配的资金有限，所以在《需求说明书》上会有所取舍。当然不管最终怎么样，为了方便阅读，我们要把开发的模块名称和内容说清楚。

《需求说明书》的范文如下。

量化投资专家系统 需求说明书

拟制人：王昭东

审核人：副总经理

批准人：总经理

2017年9月30日

一、引言

（一）编写的目的

主要就解决一个问题：我们想要做什么。

（二）背景

（1）待开发的系统的名称：量化投资专家系统。

（2）本项目是我们专门为集团量身定做的综合型量化投资专家系统，以下简称该系统。

（三）定义

无。

（四）参考资料

无。

二、任务概述

（一）目标

指导量化投资专家系统的设计、开发、测试与维护。

（二）用户的特点

该系统的用户分为三个类型：交易员、管理员和员工。其中，交易员既是专家，也是最终用户；管理员既是后台管理者，也是知识工程师；员工则是集团其他部门的在职员工，包括客服、客户经理、总裁办员工及其他分公司员工。

（三）假定和约束

无。

三、需求规定

具体功能模块详见以下表格，当然也可以绘制一个框架图，或者用文字进行描述。

数据部分的功能模块如下表所示。

模块名称	模块说明	输入	输出
数据采集	百度指数、微博舆情、官方公告、财经资讯、突发事件、IP 索引库、数据降噪、网址权重、实时更新（特殊网址要定时请求）	网址	网页内容

续表

模块名称	模块说明	输入	输出
数据接入	交易所 L2 (交易量、实时行情、H 股、A 股、恒生、沪指、道琼斯)、银行利率 (中国、中国香港、美国) 等	股票代码	股票信息
税率费率	国家税率 (中国、中国香港、美国)、债券数据及官方数据推送 (新闻、公告、股指变动)	国家机关官方网址	最新税率费率信息
数据购买	历史分笔逐笔交易 (A 股、H 股)、L2 及舆情数据等	HTTP 接口	数据信息
数据专员	数据清洗、非结构化数据结构化、数据标注	非结构化数据	结构化数据
数据库管理	大多以接口的形式进行管理, 分权治之	数据接口	数据内容
字段管理	自定义搜索字段, 需要管理员审核	查询 SQL	数据信息
统计功能	使用 JavaScript 插件绘制股票和期货行情 K 线图, 使用 echarts.js 等进行统计分析, 允许用户自定义统计插件	数据内容	统计图表

人工智能部分的功能模块如下表所示。

模块名称	模块说明	输入	输出
机器学习	有监督图片学习, 第二期包括无监督图片和文本学习	图片、文字	特征模型
知识库	使用可视化界面编写知识库	可视化操作	知识、代码
推理机	似然推理、逻辑推理、归纳推理	推理	最优结果
数据分享	交易员之间分享数据	分享数据	查看数据
词库管理	词库增、删、改、查	词库	词库
强化学习	以盈亏比等为中心进行机器人的强化学习	模拟、实盘	优化交易结果
个性设置	机器人的相关属性设置 (好奇指数、遗传指数等)	机器人	机器人
自定义函数	用户自定义函数和指标	代码	指标
沙盘推演	趋势预测、价格预测 (可以使用未来函数)	模拟投资	模拟收益

交易部分的功能模块如下表所示。

模块名称	模块说明	输入	输出
自动交易	第二期有软件自动化交易	投资	收益
模拟炒股	模拟股票交易 (实时行情)	模拟投资	模拟收益
下单系统	下单管理、撤单管理、手动下单、条件单、追涨单等	投资	收益
风控管理	强制止损、实时监控、按照不同策略的不同盈亏比进行风控管理, 参考凯利公式	投资	收益
提醒功能	邮件通知、站内信和短信通知	系统自动生成	通知内容

办公系统部分的功能模块如下表所示。

模块名称	模块说明	输入	输出
动态权限	模块管理、权限管理、角色管理、角色员工管理等	权限划分	权限
动态菜单	用户自定义菜单	自定义菜单	显示菜单

续表

模块名称	模块说明	输入	输出
员工管理	通讯录管理、组织机构管理、员工管理	员工信息和关系	显示员工信息和关系
客户管理	客户管理（客户导入、添加、修改、分配、跟踪）	客户信息	客户信息
销售服务	通话记录、客服分配、防撞单设置	销售服务日志	销售服务日志
短信邮件	邮件、站内信、通讯录	发送信息	查阅信息
仪表盘	自定义桌面，默认是操作记录和考勤信息及通知	略	略
账户管理	密码、信息、密码找回（人工方式）	略	略

运维管理部分的功能模块如下表所示。

模块名称	模块说明	输入	输出
集群管理	私有云服务器管理（IP、服务器、机柜、机房、带宽等）	启用或禁用	设备的启用或禁用
交易监控	异常交易提醒（主要是与专家系统提示相反的交易）	反向交易	交易信息
日志管理	日志的查询	系统日志	日志信息
数据备份	数据的备份与导出	数据	数据备份
系统预警	磁盘剩余、内存峰值、网络流量异常等	自动	预警信息
异地容灾	数据迁移、系统迁移、消息推送、数据恢复、系统恢复（版本控制）	暂停或恢复	系统暂停或恢复
系统测试	测试报告、单元测试、压力测试、微服务测试	测试软件	测试结果
API 管理	自定义接口（主要是数据接口和插件）	接口	接口
安全监测	SQL 注入、DRDOC、跨站脚本、CA 证书、令牌管理、Token	检测	安全信息
多账户管理	统一账户系统，集团原来是系统账户接入管理	账户	第三方系统
任务轮询	多任务并行管理	轮询	任务执行

其他模块如下表所示。

模块名称	模块说明	输入	输出
产品管理	金融产品的增加、修改、删除	略	略
财务管理	交易列表、资金的申请和划拨	略	略
插件管理	插件的添加	略	略
其他	根据需要添加的临时功能	略	略

四、运行环境规定

该系统采用 B/S 架构，服务端运行于企业私有云中的 Linux 系统上，客户端运行于浏览器（浏览器需要支持 HTML5，推荐谷歌浏览器）上。

2.4 项目开发计划

有了具体的《需求说明书》以后，就要思考这个项目到底需要多长时间能够完成，正常情况下，一个项目的具体技术管理者应该参与到“可行性研究”和“需求说明”讨论会中，并且适时给出技术上的合理建议。所谓适时，是指当需求要落实到纸上的时候，我们要及时给出一个大概的成本估量。

项目开发计划最重要的就是开发周期，也就是时间人工成本。我们建议项目开发计划的开发周期在技术部门内部评估的正常时间的基础上再乘以 1.5~3 倍。因为很多意外情况是技术部门无法预料的，若出了问题，技术部门就要承担责任。

项目开发计划范文如下。

量化投资专家系统 项目开发计划

拟制人：王昭东

审核人：副总经理

批准人：总经理

2017 年 11 月 30 日

一、引言

（一）编写目的

一方面让投资人可以了解项目开发的大概进度，做到心中有数，另一方面也方便核心技术人员确立一个基本的架构体系和指导思想，避免项目无故延期。

（二）背景

（1）项目名称：量化投资专家系统。

（2）本系统采用互联网模式为集团和投资人开发一套智能的量化交易管理系统，系统采用 PHP 7.0 作为主要开发语言，通过 SAAS 模式运行于计算机和其他智能终端上。

（3）本系统采用面向组件的模式与很多优质的第三方金融接口进行对接，从而避免系统大面积瘫痪。

（三）定义

无。

（四）参考资料

无。

二、项目概述

利用先进的计算机系统和强大的数据库进行投资交易管理，从而提高效率、保证利润、减少成本、扩大规模和发现新策略！

（一）工作内容

本系统主要工作涉及：产品原型、程序架构、核心模块、软件测试、系统集成、技术难点和备选方案。

（二）主要参加人员

首席执行官、首席技术官、系统架构师、核心技术人员、产品经理和投资人代表。

（三）产品

本项目使用以瀑布开发模式为主、以迭代开发模式为辅的双模式进行自主开发，并以 PHP 作为主要编程语言，同时以 Java、Python、JavaScript、C++等编程语言为辅。数据库则以 MongoDB 和 MySQL 为主。服务端为 Linux（Ubuntu 和 Centos）系统。客户端运行于谷歌浏览器上。

项目开发好后需要将源码、设计文档、开发文档、测试报告和使用说明书等移交给运营部门，同时提供培训服务和 Bug 修复工作。

（四）验收标准

验收标准以《需求说明书》为准，以产品原型图为辅（实际上我们是以产品原型图为主的）。

（五）项目周期

项目周期为 3 年，其中第一期为半年，第二期为 1 年，第三期为 1 年半。这个时间实际上是根据阶段性开发周期来估算的。

三、实施计划

这里可以写一个大概的实施计划，如果是一个有经验的团队，最好能够根据实际情况给出一个具体的开发计划，比如生成一个真正实用的甘特图，详细开发计划可能如下图所示。

ID	任务名称	开始时间	完成	持续时间	负责人	% 完成
1	专家系统原型	2017/2/6	2017/2/10	5天	产品：王昭东	0%
2	专家系统设计	2017/2/13	2017/2/24	10天	设计：外包	0%
3	数据库设计admin,tree,log,im,member	2017/2/13	2017/2/24	10天	DBA：王昭东	0%
4	专家系统开发	2017/3/1	2017/3/31	23天	开发：PHP高级5人	0%
5	专家系统测试	2017/4/3	2017/4/14	10天	测试小组	0%
6	专家系统模拟	2017/4/17	2017/4/28	10天	模拟：王昭东	0%
7	专家系统上线	2017/5/8	2017/5/31	18天	系统发布组	0%
8	股票数据图片化（K线）	2017/6/5	2017/6/30	20天	开发：PHP高级	0%
9	深度学习框架搭建	2017/7/3	2017/7/31	21天	架构：王昭东	0%
10	深度学习调试	2017/8/7	2017/8/31	19天	调试：DL高级	0%
11	K线图片学习	2017/9/4	2017/10/31	42天	学习：DL高级	0%
12	策略修正	2017/11/1	2017/11/30	22天	策略审核组	0%
13	策略模拟	2017/12/1	2017/12/29	21天	模拟：王昭东	0%

关键问题如下。

（1）设计延期：设计两套方案，一套是基本设计方案，另一套是相对比较优秀的方案，如果相对比较优秀的方案没有能够按时完成，则使

用基本设计方案进行开发，注意 MVC 架构。

(2) 数据库不实用：架构师要积极参与到整个项目的讨论中，对于产品的需求从数据库的角度进行思考，如果无法完成数据库，则及时提出意见。同时采用双 DBA 的策略，让两个数据库负责人可以相互讨论，但是必须以其中一个人为主。数据库经过架构师审核后上线，数据库的设计文档最好附有增、删、改、查的案例。

(3) 程序开发延期：在项目开发前做好计划，将模块细化，精确到工作日或工作周，尽量不以自然月为单位。同时对整个任务周期乘以 1.5 倍作为项目缓冲期。项目模块化分工的时间可以略长一些并邀请有经验的程序员参与，避免程序开发好以后模块之间衔接不上。

四、支持条件

略。

2.5 产品原型图

作为一个实际的项目，技术是需要围绕产品的改变而改变的，而一般的产品经理很难把握未来一年内的市场趋势，虽然量化投资专家系统相对成熟稳定、功能相对固化，但这并不表示不需要增加或修改既定的功能，因为在一年的时间里，公司很可能增加一个部门或成立一个特殊业务的子公司。这对于产品经理来说，无疑是一个很大的挑战，因为很难预测公司具体业务的发展规划，所以就很有可能修改原来定好的需求说明。修改需求说明是一件非常麻烦的事情，如果是外包项目，就会产生很多额外的预算，甚至会使整个外包项目都可能无限期顺延甚至失败。如果由自有团队来做，技术总监同样也会非常害怕修改需求说明。

为了解决需求不能被修改与不修改体验差的两难问题，我们建议一方面需要产品经理或技术总监非常有经验、有先见之明，提前做好规划、减少改动，另一方面技术总监采用迭代开发的方式，尽可能合理地留出一些

扩展，并做好“拥抱”变化的准备。

折中的办法是自有团队以《需求说明书》为辅，以产品原型为主，但是原型图的修改和版本要保留，所有修改的责任在于产品总监，并不在于技术总监，技术总监在给出建议的同时需要服从产品总监的修改决定，保证工期。

如果是外包团队，那么产品总监不断修改《需求说明书》几乎是不现实的，因为这个成本太高了。因此，以下项目的修改是以自有团队为前提的。

1. 产品原型

Axure RP 是一个专业的产品原型设计工具。在国内的 Web 设计工具中占有很高地位，可以说是产品经理的一个必备工具。产品经理之所以喜欢 Axure RP 主要得益于它的可视化操作、灵活的 JavaScript 脚本和一键生成 HTML 页面的功能。Axure RP 7.0 界面如图 2-2 所示。

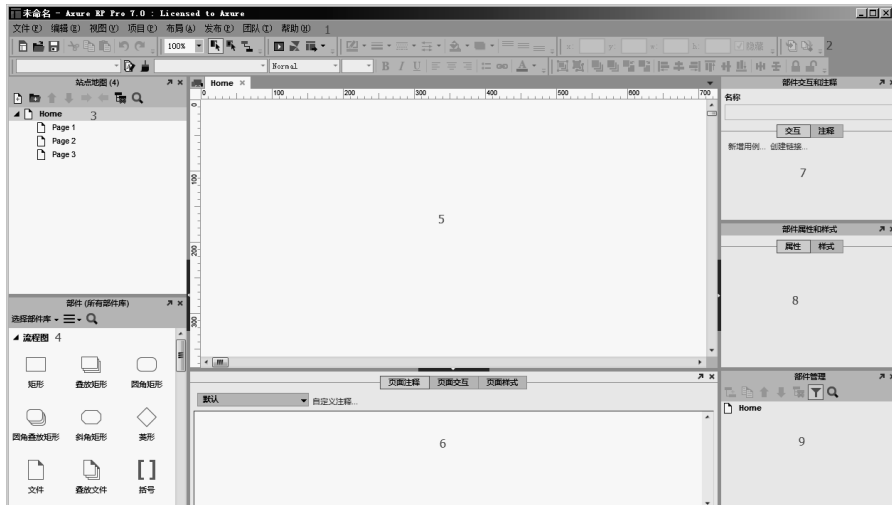


图 2-2 Axure RP 7.0 界面

Axure RP 7.0 界面功能如下。

图 2-2 中的区域 1 是菜单栏，包括文件、编辑、视图、项目、布局、

发布、团队和帮助。其中，【文件】菜单主要包括新建、打开、保存和打印页面功能；【编辑】菜单主要包括粘贴、复制、查找、替换、撤销功能；【视图】菜单主要是控制主界面的面板和工具栏；【项目】菜单不太常用，主要是进行与样式、全局变量相关的设置；【布局】菜单主要有对齐、分布、图层、组合等功能；【发布】菜单最常用的功能就是预览，预览的快捷键是F5键，此外还可以生成说明书和原型文件；【团队】菜单和【帮助】菜单一般项目用不到。

图 2-2 中的区域 2 是常用工具按钮，包括新建、打开、保存、预览等。

图 2-2 中的区域 3 是站点地图，通过站点地图可以看到站点的目录结构，双击某一个节点可以快速进入这个页面。

图 2-2 中的区域 4 是部件库，常用按钮、图形、控件全在这里，只要将其拖曳到绘制中心（区域 5）即可。

图 2-2 中的区域 5 是页面的绘制中心。

图 2-2 中的区域 6 是页面注释、交互与样式面板，通过这些面板既可以对页面进行说明，也可以对具体的按钮添加事件，同时还可以修改相关样式。

图 2-2 中的区域 7 和区域 8 与区域 6 相互关联，当选中一个按钮时才会显示出这个按钮所具备的交互功能、注释、属性和样式。

图 2-2 中的区域 9 是部件管理，一般不常用。

Axure RP 这款软件最大的特点就是上手容易，只要把想要的部件从部件库里拖到对应的绘图区域就可以了，如果还想做得更真实一些，就再添加一些交互事件，比如超链接。

原型图工具有了，最重要的是一个网站的设计思想，比如页面大小、元素布局和功能要求。要想做出一个好的原型图，80%的时间都要用在思考上。

例如，量化投资专家系统该有哪些功能，这些功能具体应该怎样实现等，这些都是产品经理要考虑的事情。

通常，我们会以用户角色扮演的方式来进行设计，比如管理员应该看到什么、员工应该看到什么、交易员又应该看到什么，然后把他们共有的部分做成一个公用的模块。

这样划分后，就可以把整个量化投资专家系统划分为登录、专家系统、管理员、统计分析、消息任务、账户管理、组织机构和常用窗口几个部分。

例如，登录模块可以使用下拉列表的形式，做成让不同用户进行选择



图 2-3 登录模块

其中，图 2-3 中所标注的 1、2、3 是交互事件，交互事件必须在预览之后才能生效，这些交互事件主要是一些跳转按钮，比如单击【忘记密码】按钮后打开忘记密码页面，单击【登录】按钮后打开默认首页等。

2. 专家系统

专家系统是一个大模块，这个大模块下面又包含很多小页面，这些小页面又有一个共同的菜单。共同的菜单可以使用母版的方式来生成，这样只要修改母版里面的内容，所有调用这个母版的菜单就一次性都修改好了。另外，专家系统默认是一个列表页，在这个页面里面添加了高级搜索和分页符。列表页如图 2-4 所示。

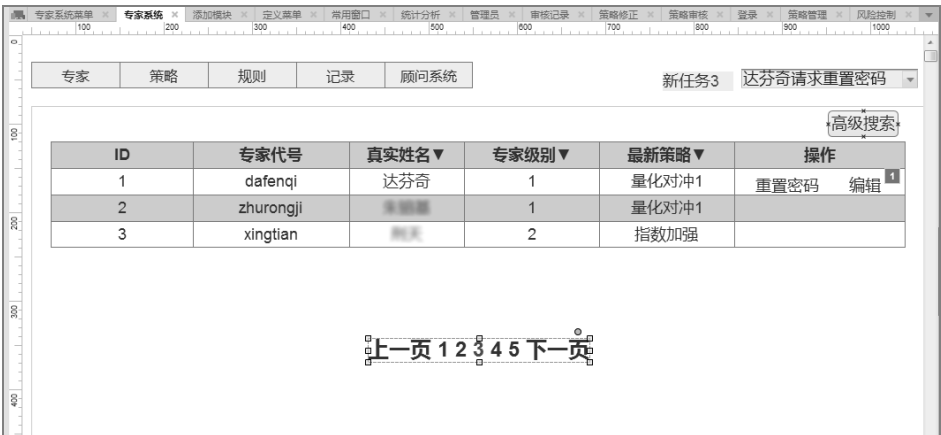


图 2-4 列表页

专家系统模块包含专家、策略、规则、记录和顾问共 5 个子模块，这些子模块之间的关系如图 2-5 所示。

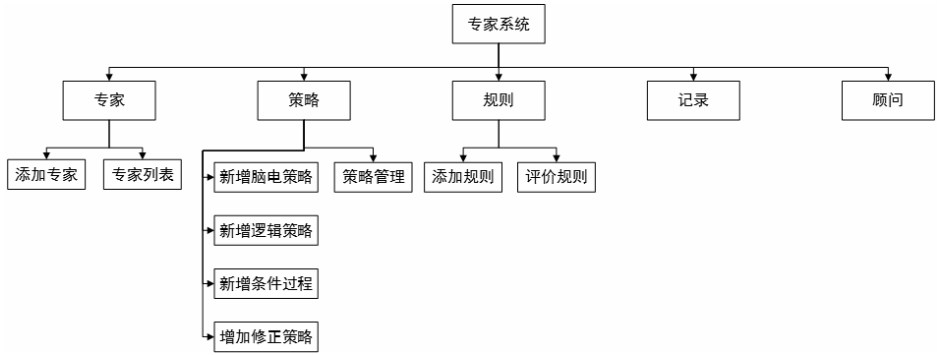


图 2-5 专家系统子模块之间的关系

3. 添加专家

添加专家时，一般需要添加专家的个人信息和专家级别，如果基本信息不够，管理员还可以通过【+】按钮添加自定义字段。如果觉得整个表单下次还有用，则可以单击【存为模板】按钮存储当前设计好的表单。已有的表单模板可以通过【默认模板】下拉菜单进行选择。添加专家页面如图 2-6 所示。

照片 默认模板

姓名 陈明部

性别 无

生日 1983-12-12 23:30

学历 无

电话 010-8888888

住址 北京·王府井大街

血型 无

入职 2016-12-12 23:30

简介

专家级别 0 ☒ 禁用 ☐ 启用

图 2-6 添加专家页面

如果只想添加会员，而不添加那么多个人信息，则可以单击图 2-6 中的【只添加会员】按钮，进入会员添加页面，如图 2-7 所示。

专家

密码

通知方式

图 2-7 会员添加页面

4. 专家列表

专家列表页和专家系统默认页是同一个页面，如图 2-4 所示。专家列表主要包括专家的 ID、代号、真实姓名、专家级别和该专家最新的策略，同时为了方便浏览，还可以通过真实姓名、专家级别和最新策略进行排序。单击【重置密码】按钮可以重置这个专家的登录密码，单击【编辑】按钮可以修改这个专家的个人信息。

专家列表中还有一个【高级搜索】按钮，在专家数量很多时可以进行高级搜索。

5. 新增条件策略

在正常情况下，最好给交易员一个可视化的操作界面，通过一些选择就可以完成一个相对简单的策略。新增条件策略就是这样一个可视化操作界面，由于股票和期货有所不同，所以把股票和期货分别做成了两个页面，股票页面如图 2-8 所示。

图 2-8 股票新增条件策略页面

期货页面如图 2-9 所示。

图 2-9 期货新增条件策略页面

当然还可以更加精简，只要根据价格进行条件下单就好。一般为了方便条件下单，还可以提供一些经典的策略供交易员修改，比如波浪线、折线、射线等，如图 2-10 所示。

期货	sr1701	月	未来一年白糖预测	线形	波浪	射线	折线	上弧	下弧	通过	自定义
2016-12	开盘	100	收盘	100	最高	100	100	100	100	交易量	100
2016-12	开盘	100	收盘	100	最高	100	最低	100	100	交易量	100
2016-12	开盘	100	收盘	100	最高	100	最低	100	100	交易量	100
2016-12	开盘	100	收盘	100	最高	100	最低	100	100	交易量	100

图 2-10 经典策略页面

6. 新增逻辑策略

一般掌管较多资金的交易员的策略都是逻辑上非常清晰的组合策略，其中股票池、策略池、决策树、回测、模拟等都是经过严格验证的，这个功能在 JavaScript 交互部分较多，如图 2-11 所示。

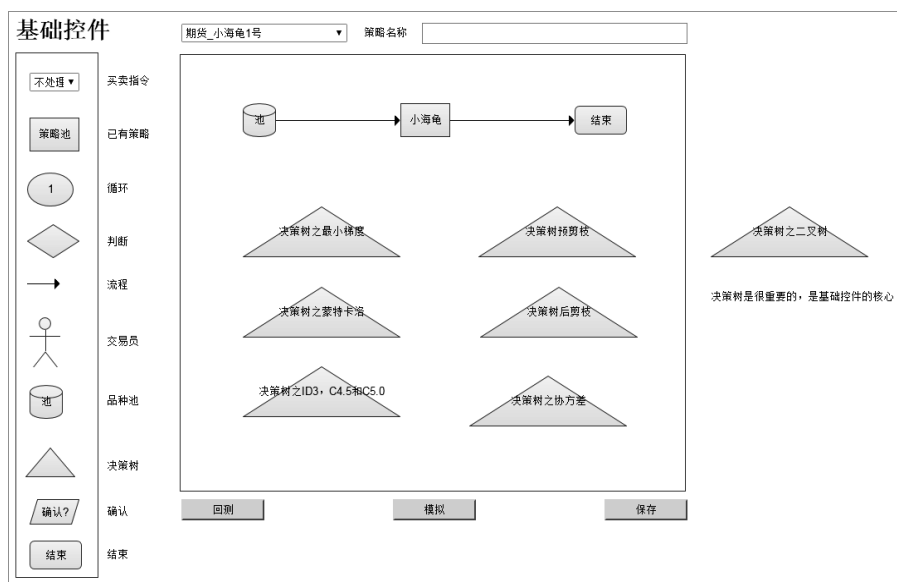


图 2-11 逻辑策略页面

7. 新增脑电策略

由于很多优秀的交易员或金融分析师并不能很好地总结出策略思想，

只有一种感觉，所以需要借助脑电波控制仪、摄像头和屏幕录制软件对其所进行的操作（包括最终决策）进行机器学习，从而找出这位交易员的策略思想。其中，机器学习后的交易指令过程如图 2-12 所示。

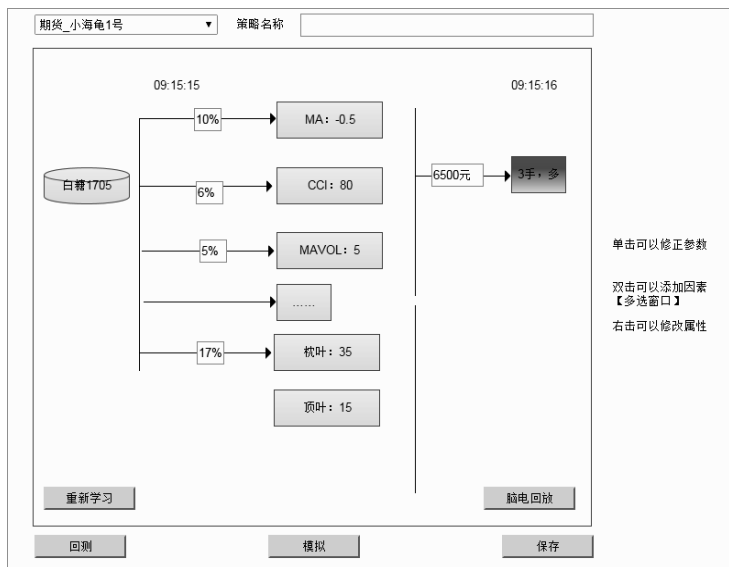


图 2-12 机器学习后的交易指令过程

图像化演示部分的脑电、屏幕等录制过程如图 2-13 所示。

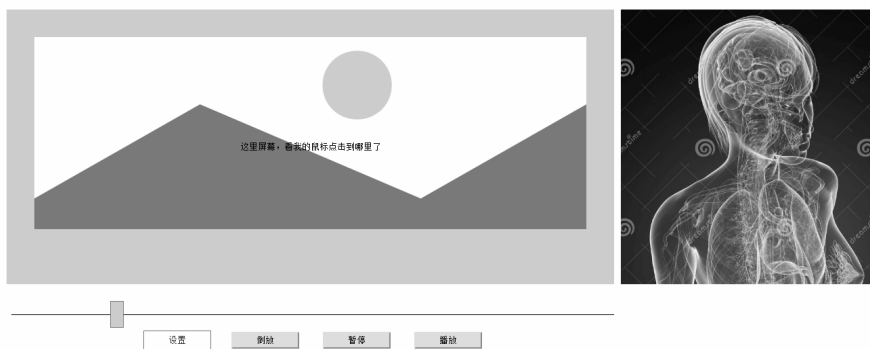


图 2-13 录制过程

8. 新增修正策略

修正策略是一个很重要的功能，它是一个真正意义上的 Web 编译器(编

程语言开发环境)。新增条件策略、新增逻辑策略、新增脑电策略最终都会变成计算机可以理解并执行的代码,为了更好地完善我们的策略,需要一个能够更细致地修改代码的页面,其中第一期应该开发基本的源码编辑功能,第二期需要完善一些提高工作效率的功能,如图2-14所示。

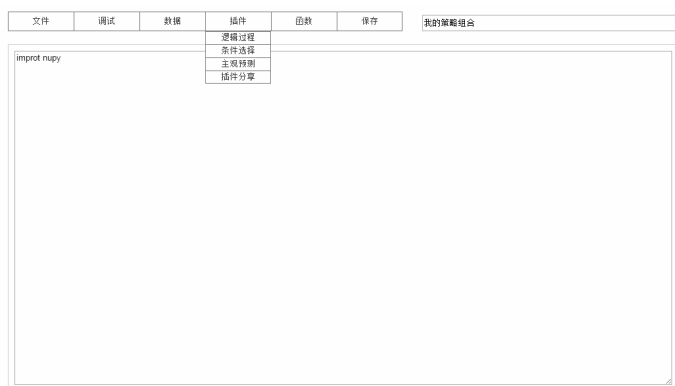


图 2-14 修正策略页面

9. 添加规则

添加规则非常简单,就是一个文章编辑页面,主要是为策略审核提供一个参考的依据,避免审核员过于主观地推荐策略上线,如图2-15所示。

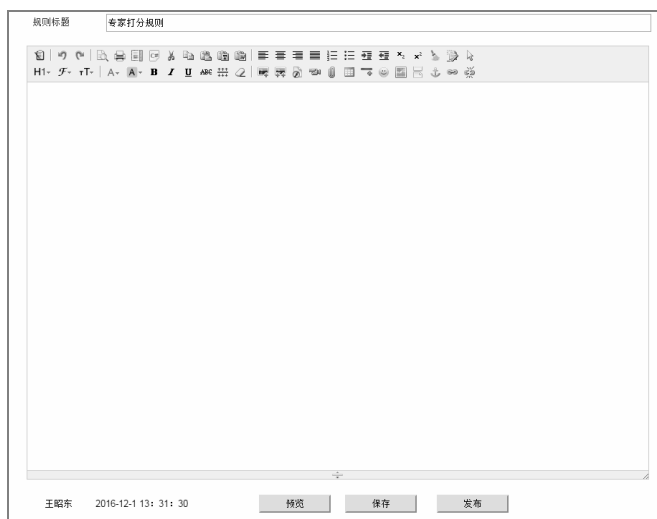


图 2-15 规则页面

10. 评价规则

评价规则页面就是已经添加规则的文章列表页，由于比较简单，所以类似的列表页就不展示了。

11. 记录

这里主要是交易的模拟记录，方便交易员、审核员直观地感受策略的运行效果，如图 2-16 所示。

策略思想	源码	测试	操作
来源于海龟策略，短平快	PYTHON源码 附件	<input type="button" value="回测"/> <input type="button" value="模拟"/>	权重: 0 <input type="text"/> <input type="button" value="同意"/> <input type="button" value="不同意"/>
来源于海龟策略，短平快	PYTHON源码 附件	<input type="button" value="回测"/> <input type="button" value="模拟"/>	权重: 0 <input type="text"/> <input type="button" value="同意"/> <input type="button" value="不同意"/>

图 2-16 记录页面

单击图 2-16 中的【回测】、【模拟】、【明细】等按钮可以看到这个交易策略的具体表现，如图 2-17 所示。

概述	统计	日志	视图	添加	日期	2016-12-1	—	2016-12-31						
2016-12-1 — 2016-12-31					合约可以用英文半角逗号分割: <input type="text"/>				<input type="button" value="搜索"/>					
合约▼	多空▼	盈亏▼	交易次数▼	手数▼	最大回撤▼	胜率▼	盈利率▼	最大亏损▼	开始时间▼	结束时间▼	交易结果▼	平均时长▼	平均利润▼	利润▼
											盈亏			
											平仓			
											平仓			
小计:												15分钟	500元	15000元
上一页 1 2 3 4 5 下一页														

图 2-17 交易策略的具体表现页面

12. 投顾系统

投顾系统更多地是为集团的大型投资并购业务服务的，主要在一级市场，而非二级市场。

13. 管理员

管理员页面是一个很重要的模块，也是整个系统的初始模块，在管理员页面可以管理员工、分配权限、修改模块、添加插件、查看日志和维护系统。管理员页面的功能框架如图 2-18 所示。

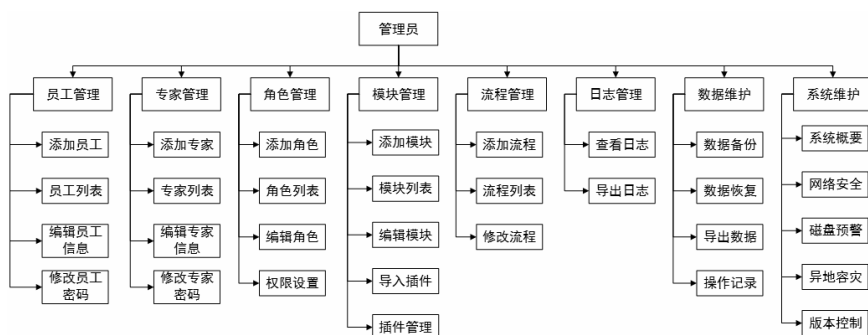


图 2-18 管理员页面的功能框架

14. 员工管理

员工管理页面就是一个简单的列表页，通过列表页可以看到员工的账户名称、真实姓名、登录的时间及登录的 IP 地址。单击列表页中的【重置密码】按钮可以修改该员工的密码，单击【编辑】按钮可以编辑该员工的基本信息，如图 2-19 所示。

ID	账户名称	真实姓名	上次登录地址	上次登录时间	操作
10	dafengqi	达芬奇	北京	2016-12-11	重置密码 编辑
20	zhurongji	朱德基	北京	2016-12-11	
25	xingtian	刑天	北京	2016-12-1	

新任务3 达芬奇请求重置密码 专家 任正非 搜索

员工管理 专家管理 角色管理 组织机构 模块管理 日志管理 数据维护 系统维护 流程管理 统计分析 消息 账户

上一页 1 2 3 4 5 下一页

图 2-19 员工管理页面

15. 专家管理

专家管理页面和员工管理页面类似，这里不再列举。

16. 模块管理

本系统的所有权限都是通过模块来确定的，模块是权限的最小单元，

主要包括名称、简介、排序、URL、Token 和函数集，其中 URL、Token 和函数集是不能写错的，否则系统将无法调用这个模块或模块下的具体函数。添加模块页面如图 2-20 所示。

图 2-20 添加模块页面

编辑模块页面和添加模块页面非常相似，这里就不展示了。

17. 添加插件

插件是模块的一个综合体，很多时候需要在系统上留出一定的扩展，以满足意外需求，这时候就可以通过插件的形式把单独开发好的程序上传到插件目录里。插件的上传一共分为三步，第一步是客户端上传文件，如图 2-21 所示。

图 2-21 客户端上传文件页面

第二步是配置插件，很多插件都需要连接到指定的数据库中，如图 2-22 所示。



配置插件

数据库配置 IP 账户 密码

页面初始大小 高 像素 宽 像素

1.导入文件》2.配置插件》3.管理插件

下一步

图 2-22 配置插件页面

第三步就是插件管理，对已经添加的插件决定是否启用，如果配置文件出现了改动，在这里就可以进行修改，而不用重新上传插件，如图 2-23 所示。



插件管理

名称	作者	上级	状态	时间	操作
扇形统计图插件	比尔盖茨	统计	禁用	2016-12-11	修改配置

1.导入文件》2.配置插件》3.管理插件

体验 保存

图 2-23 插件管理页面

这里还有一个需要说明的地方，就是插件上传完之后并不能马上使用，即便启用了也不行，因为只是上传了源码文件，系统并不知道这个源码文件怎么该调用，这时候还要在模块管理中添加新上传的具体模块。

18. 角色管理

添加角色（与编辑角色相似）是角色管理中的一个核心功能，在添加一个角色时，需要根据用户类型进行权限选择，管理员的权限和员工的权限是分开的，为了方便管理员编辑权限，我们把已有的模块都列了出来并加以区分，管理员只需要勾选对应的模块即可，如图 2-24 所示。

图 2-24 添加角色页面

19. 权限设置

权限设置就更简单了，找到指定的员工，然后选择一个对应的角色，单击【确定】按钮即可，如图 2-25 所示。

图 2-25 权限设置页面

20. 流程管理

为了适应集团不同的业务需求，我们需要一个动态的流程管理系统，这个系统指明什么人拥有使用什么模块的权限，然后在其使用后通知给下一个人，下一个人接收到通知后选择同意或不同意，直到结束。下面以添加流程为例进行讲解。

我们需要设置流程的名称，指定员工或某个角色下的所有员工（一般员工都对应着权限），当这个员工在数据库中成功增加一条记录（表示完成了一个申请）时，系统就以消息的形式发送给审批者，审批者可以是直接

上级，也可以是某个具体的员工或全部下属。如果审批者是多人，则可以设置通过率来决定这个申请是否通过。同时为了防止延期，还要增加一个时限的字段，比如两天。

以上仅仅是动态流程中的一个环节，动态流程可能会有很多环节，这时可以通过【+】按钮来增加下一个环节，如图 2-26 所示。

图 2-26 动态流程页面

单击图 2-26 中的【预览】按钮，会弹出流程辅助测试页面，通过预览可以查看流程是否有漏洞，如图 2-27 所示。

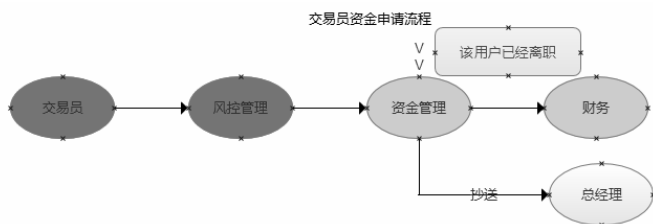


图 2-27 流程辅助测试页面

21. 数据接口

在数据维护中，数据接口是一个关键点，不管是维护自己的数据库，还是连接第三方的数据库，都不希望反复输入账户名和密码，特别强调安全的数据库除外，这时候就可以做一个数据接口来保存所有的账户名和密码。一

般数据接口主要包括地址、账户名和密码这 3 项，如图 2-28 所示。

数据接口 新数据接口

名称

地址

账户

密码

token

费用

日期 2016-12-1 — 2016-12-31

联系人

代码 浏览

说明文档 浏览

文件类型 SQL

表头位置 第 1 行

映射

文件	格式化	目的地表(gupiao)
名称	转数字	名称 (sting, ▼)
数值		

修改表

范文：
交易新L2 (交易量，实时行情，H股，A股，恒生，沪指，道琼斯)
银行利率 (中国，香港，英国)
货币汇率 (中国，中美，港美)
国家航单 (中国，香港)
债券数据
数据推送 (新闻，公告，股指期货)

保存

图 2-28 数据接口页面

22. 导入数据

导入数据时一般需要做一个简单的字段映射，如果数据库中没有对应的表，那么系统会根据字段直接生成一个表，字段默认是字符串类型。为了安全起见，只能从默认数据库导入、导出数据。如图 2-29 所示为导入数据页面。

导入数据

文件类型 SQL

目的地表 gupiao 映射

文件地址 浏览

表头位置 第 1 行

映射

文件	格式化	目的地表(gupiao)
名称	转数字	名称 (sting, ▼)
数值		

修改表

开始

修改表

表名称 gupiao

ID id int 自增长 10

名称 name str 无 20

重置 另存 保存

图 2-29 导入数据页面

23. 导出数据

导出数据页面如图 2-30 所示。

导出数据

文件类型

SQL

导出位置

浏览

表名

name_log_01

选择表

检索

表名	大小	创建时间	备注
<input type="checkbox"/> name	10K	2016-12-20	用户表
<input type="checkbox"/> log_01	100K	2017-12-1	日志表

开始

图 2-30 导出数据页面

24. 系统维护

系统维护页面更像是一个仪表盘，方便管理员了解当前系统运行的基本情况，比如请求量、剩余空间和系统日志等，如图 2-31 所示。



我们还可以做一些与预警相关的设置，这样就不用总盯着屏幕工作了，比如磁盘预警设置，如图 2-32 所示。

预警设置

☐ 剩余空间

☒ 使用率

☐ 日增速

☒ 瞬间增速

☒ 通知方式

保存

图 2-32 预警设置页面

25. 异地容灾

异地容灾页面相对简单，主要就是备份和恢复数据，由于工程项目较大，所以可以使用第三方工具，如图 2-33 所示。

备份至

☐ 香港地区
 ☒ 成都云中心
 ☐ 刻录成盘

从何处恢复

☒ 成都云中心
 ☐ 香港地区

80%

备份数据

80%

恢复数据

图 2-33 异地容灾页面

26. 统计分析

统计分析也是一个非常重要的大模块，这个模块从一定意义上说就是一个简单辅助决策系统，统计分析的框架如图 2-34 所示。

27. 概述

概述页面也是默认页面，主要介绍某个策略指定周期内大概的交易情

况，比如累计收益、年化指标、最大回撤等，如图 2-35 所示。

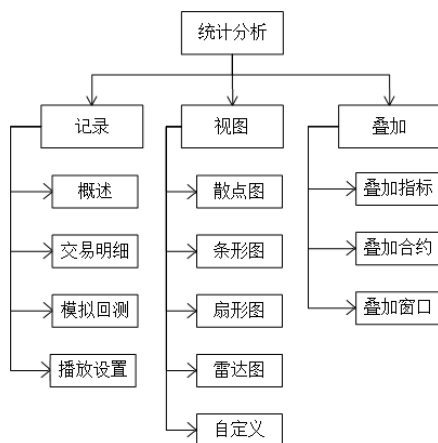


图 2-34 统计分析的框架



图 2-35 概述页面

28. 播放设置

如果对播放的速度、日期等特殊要求，可以单击【设置】按钮进行相关的设置，如图 2-36 所示。

29. 交易明细

交易明细页面与成交记录页面相似，此处不再赘述。

帧频

1

秒/帧

1

数/帧

保存

范围

日期

2016-12-1

—

2016-12-31

价格

1000

—

1500

保存

次数

1

次

保存

图 2-36 播放设置页面

30. 统计视图

为了方便展示统计的结果，可以做几个直观的统计图，比如散点图、条形图、折线图等。条形图如图 2-37 所示，其他统计图类似。

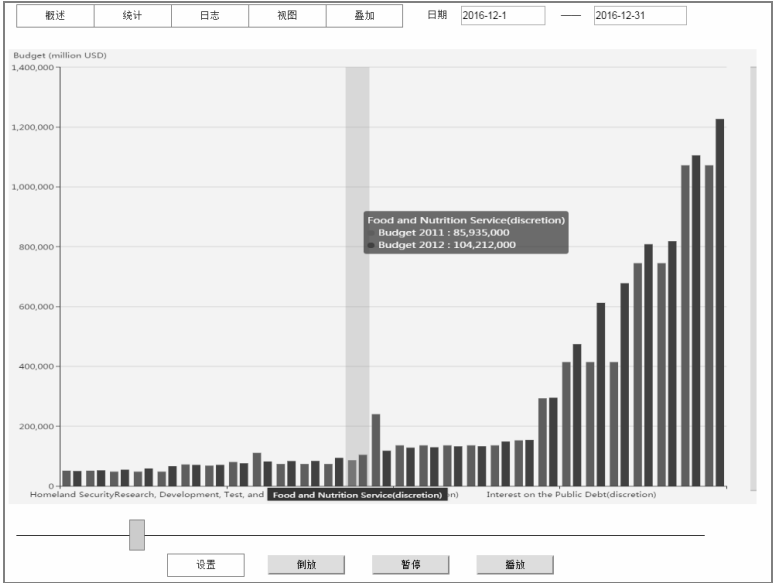


图 2-37 条形图

31. 消息任务

消息任务模块和正常的邮件类似，单击【+】按钮可以选择用户，如图 2-38 所示。

该图展示了消息任务页面的界面。顶部有一个标签页，包含“邮件(3)”、“任务”、“消息”、“设置”和“已发送”，其中“消息”标签页被选中。右侧有一个“全部”下拉菜单、一个搜索框和“搜索”按钮。主内容区域包含“收件人”和“抄送”两个输入框，均填写了“达芬奇, 王昭东, 荆天”，右侧有“+”号按钮。下方有“密抄”和“公开”两个单选按钮，当前选中“公开”。再下方是“标题”输入框，填写了“国庆节放假通知！”。接着是“正文”区域，包含“列表”、“图片”、“附件”和“格式”四个子选项卡，当前显示为一个大的文本输入框。底部有“常用内容”和“存为常用内容”两个按钮。最底部有“使用签名”和“不使用签名”两个单选按钮，当前选中“使用签名”，以及“保存”和“发送”两个按钮。

图 2-38 消息任务页面

同时，消息任务也支持自定义设置，比如签名、多邮箱管理和常用内容管理等，如图 2-39~图 2-41 所示。

该图展示了邮件签名页面的界面。窗口标题为“邮件签名”，右上角有关闭按钮“X”。内部有一个大的文本输入框，其中预置了签名模板“%date%王昭东”。底部有一个“确定”按钮。

图 2-39 邮件签名页面

该图展示了邮件收发页面的界面。窗口标题为“邮件收发”，右上角有关闭按钮“X”。顶部有一个下拉菜单，显示“295392870@qq.com”。下方包含 SMTP、POP3 和账户的输入框，均填写了“smtp://mail.qq.com”，以及一个密码输入框，显示为“*****”。底部有一个“开启SSL连接”的复选框，当前未勾选。最底部有一个“确定”按钮。

图 2-40 邮件收发页面

32. 账户管理

在账户管理页面中，用户可以修改并完善自己的信息，这个修改过程

会被日志记录，同时为了方便集团内部交流，还可以生成一个二维码名片，如图 2-42 所示。



图 2-41 常用内容管理页面



图 2-42 账户管理页面

33. 绑定账号

个人账户页面还有很多与员工切身利益相关的功能，比如为了方便密码找回和多账户登录，系统还增加了一个绑定账号的功能，绑定账号页面如图 2-43 所示。

除绑定账号的功能外，还有考勤功能，目前考勤功能是导入考勤机的数据，而不是只有系统登录的时间。

除此之外，为了方便用户管理自己的模块，还可以通过自定义菜单把

这些模块分门别类地进行管理，自定义菜单页面如图 2-44 所示。

绑定账号

账户类型 电子邮箱

是否禁用 用于登录

账户名

密码

测试 绑定

cenzie@sina.com

18519145311

15810061655

图 2-43 绑定账号页面

菜单图标

上传

上级菜单 员工

菜单名称 策略部

菜单权限

常规权限模块

消息列表

发送邮件

邮件列表

角色管理模块

添加角色

角色列表

角色设置

通讯录模块

组织机构

添加机构

常用联系人

内部联系人

外部联系人

添加联系人

我的群组

员工管理模块

添加员工

列表列表

员工编辑

禁用

启用

保存

图 2-44 自定义菜单页面

34. 组织机构

专家系统另一个比较常用的功能就是组织机构了，默认的企业通讯录就是一个组织机构的形式，同时为了符合企业的实际需要，还设置了一些不对外的私密部门，比如 X 实验室。组织机构添加页面如图 2-45 所示。

部门名称 策略部

上级部门 量化事业群

所属员工 达奇奇, 王昭东, 荆天

部门负责人 达奇奇

达奇奇

私密部门

普通部门

测试 保存

图 2-45 组织机构添加页面

• 51 •

其实专家系统还有很多原型图，但是大多数都是一些比较雷同的列表和表单，在这些页面中字段是很重要的，一个也不能少，不能因为节省时间而忽略某些字段。另外，由于产品中的字段和学术上的字段存在差异（产品是为了用户体验而使用的人性化命名），所以字段最好有专业的说明。如果没有说明，则要由产品负责人对开发人员进行培训和讲解，从而避免歧义。

最后，要加强产品原型图的版本控制，不能在原型图还没有定稿的时候就交给研发团队。

2.6 模块开发说明

产品原型是为了让研发人员知道我们要做什么，但是怎么做不是产品原型的事情，而是技术的事情，一般这个工作都是由架构师和核心开发人员共同决定的。为了条理清楚、主次分明，一般会按照几个大模块进行详细的划分，然后分别提出这几个模块的实现思路。另外，对于难以实现的小模块也要进行特殊说明，防止出现开发短板。

模块开发说明的范文如下。

量化投资专家系统 核心模块开发说明

拟制人：王昭东

审核人：副总经理

批准人：总经理

2017年12月30日

一、模块开发说明

（一）模块

量化投资专家系统的9大模块如下。

1. 后台管理员模块——后台
2. 账户管理模块——账户
3. 消息任务模块——消息
4. 统计分析模块——分析
5. 专家系统之知识库模块——知识库
6. 专家系统之脑电模块——脑电
7. 专家系统之模拟测试模块——模拟
8. 专家系统之推理机模块——推理机
9. 专家系统之解释器模块——解释器

(二) 人员

核心开发人员有15人(计划时可能比这个人数还要多,本书出于实际考虑,人员上精简了很多),默认语言为PHP,使用B/S结构。

1. 后台1人
2. 账户、消息1人
3. 分析、模拟2人
4. 知识库、推理机3人
5. 解释器1人
6. 脑电(Arduino)2人
7. 策略专家1人(兼职)
8. 数学顾问1人(兼职)
9. 设计师1人(兼职)
10. DBA1人(兼职)
11. 项目经理1人

二、模块开发情况表

（一）开发计划

为了能够尽快让项目上线，我们把整个专家系统分成了 9 个不同的模块，其中有些模块工作量比较大，需要多人参与，比如脑电模块；有些模块工作量比较小，比如消息模块，一个人就可以做几个这样的模块。

模块中的技术难点主要在推理机、知识库和脑电 3 个模块上，所以这 3 个模块的开发人员要有一定的开发经验和缜密的逻辑推理能力。

模拟和测试模块的工作人员及后台开发人员则需要具有较强的沟通能力。

同时，为了方便开发、测试和维护，需要制定一些基本的开发规范，让前后台程序员有法可依。开发规范应该遵循以下 3 点（特殊情况需要专门说明）。

- （1）全部采用 UTF-8 编码，从而避免乱码现象。
- （2）一定要添加注释，注释最起码包含说明和作者，方便维护。
- （3）在命名环节尽量减少冗余，全部用英文，避免使用中文或拼音。

（二）PHP 开发规范

为了追求开发速度，本系统采用自主开发框架，使用 MV（模型—视图）结构，遵循常规命名规范。

1. 字符编码

所有程序均使用 UTF-8 编码，禁止使用 GBK 和 GB2312 编码。

2. 目录和文件

目录使用小写字母加下画线。

类库、函数文件统一以 .php 为后缀。

类的文件名均以命名空间定义，并且命名空间的路径和类库文件所

在的路径一致。

类文件以驼峰法命名（首字母大写），其他文件以小写字母加下画线的方式命名。

类名和类文件名保持一致，统一以驼峰法命名（首字母大写）。

3. 函数和类、属性命名

类的命名采用驼峰法（首字母大写），例如 `User`、`UserType`，默认不需要添加后缀，例如 `UserController` 应该直接命名为 `User`。

函数的命名使用小写字母加下画线（小写字母开头）的方式，例如 `get_client_ip`。

方法的命名使用驼峰法（首字母小写），例如 `getUserName`。

属性的命名使用驼峰法（首字母小写），例如 `tableName`、`instance`。

以双下画线“__”开头的函数或方法作为魔法方法，例如 `__call` 和 `__autoload`。

4. 常量和配置

常量以大写字母加下画线的方式命名。

5. 数据表和字段

数据表和字段采用小写字母加下画线的方式命名，注意字段名不要以下画线开头，例如 `ai_user` 表和 `user_name` 字段，不建议使用驼峰法和中文作为数据表字段的名称。

数据库默认编码为 UTF-8。

（三）JavaScript 开发规范

为了加快开发速度，本系统使用 `jquery3.1` 和 `echar.js` 作为主要的 JavaScript 框架，其中 `jquery3.1` 用于客户端的交互，`echar.js` 用于图形图像的绘制。

为了避免与原系统的函数冲突，自定义的函数和属性使用 d 加下画线作为前缀，如 d_myFunction()。

字符编码为 UTF-8 格式。

(四) HTML_CSS 开发规范

字符编码为 UTF-8 格式。

1. HTML 代码分区

在页面的每一个功能分区前后加入注释标记，说明此区块的功能指向，例如：

```
<!--统一头开始/开始 -->
```

```
<!--统一头结束/结束 -->
```

2. 关于拼音

可以使用拼音，但是要明确，要么统一为拼音，要么默认为英文。如果使用拼音，则要有对应的字典以方便查找。

3. 文件名和目录

每个目录中都应该包含一个默认的 HTML 文件，文件名统一用 index.html。

文件名称统一用小写的英文字母、数字和下画线的组合。

文件命名的指导思想是使得自己和工作组的每一个成员能够方便地理解每一个文件的意义。

4. 图片的命名原则

(1) 名称分为头尾两部分，用下画线隔开。

(2) 头部表示此图片的大类，例如广告、标志、菜单和按钮等。

5. 内部样式

所有样式均需要外部调用，禁止编写内部样式，特殊情况需要写日志备忘。

6. 源码压缩

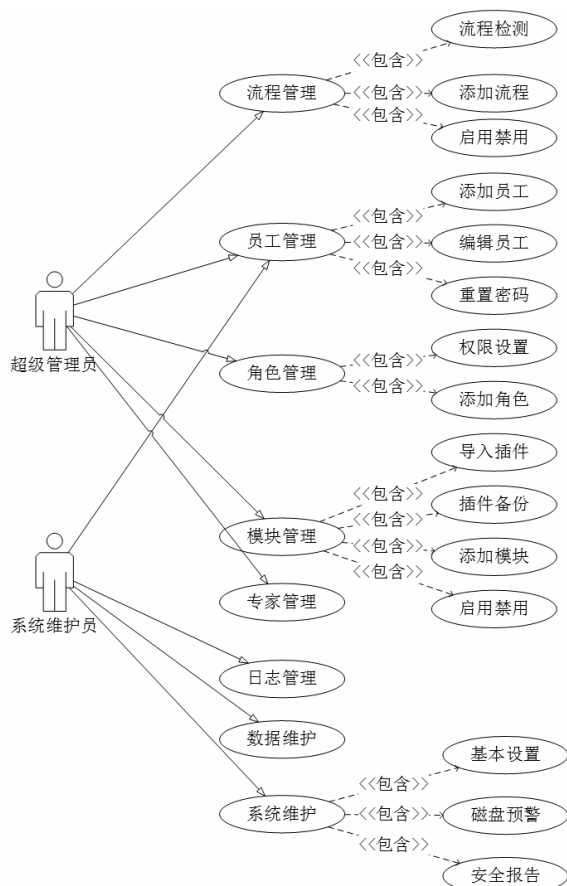
上线后才能对源文件进行压缩，开发期间的源码不允许压缩，统一使用 DW 的源码格式化。

三、系统主要模块

关于系统主要模块，我们主要给出一些流程图，不过此处所说的流程图并不是论文中的那种呆板的流程图，这里的流程图更多的是从企业实际应用角度考虑的“非正规”流程图。

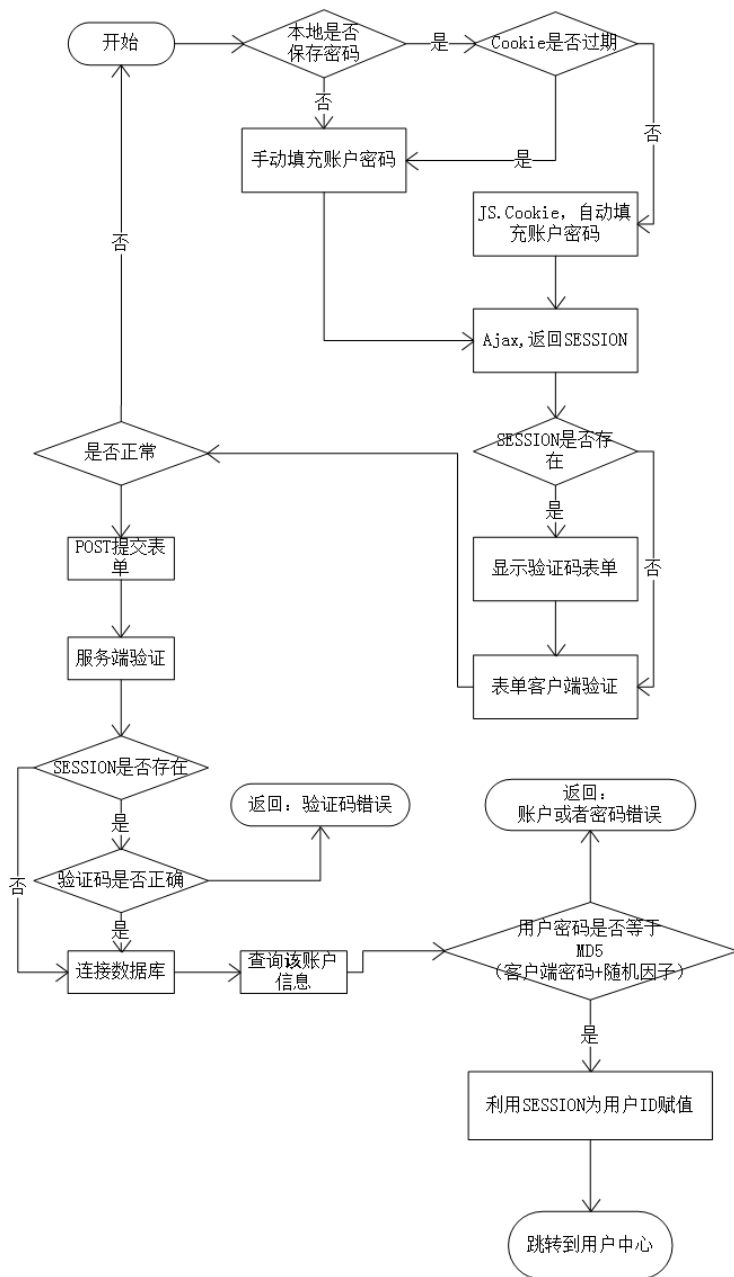
(一) 后台

管理员后台用例图如下图所示。



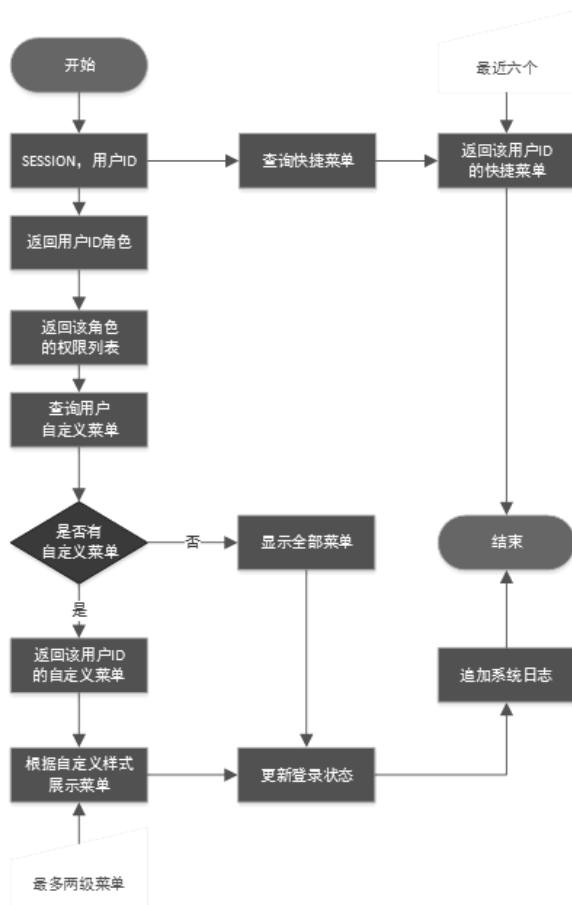
1. 登录模块

登录模块流程图如下图所示。



2. 权限菜单模块

根据角色权限调用用户的自定义菜单和快捷菜单，流程图如下图所示。



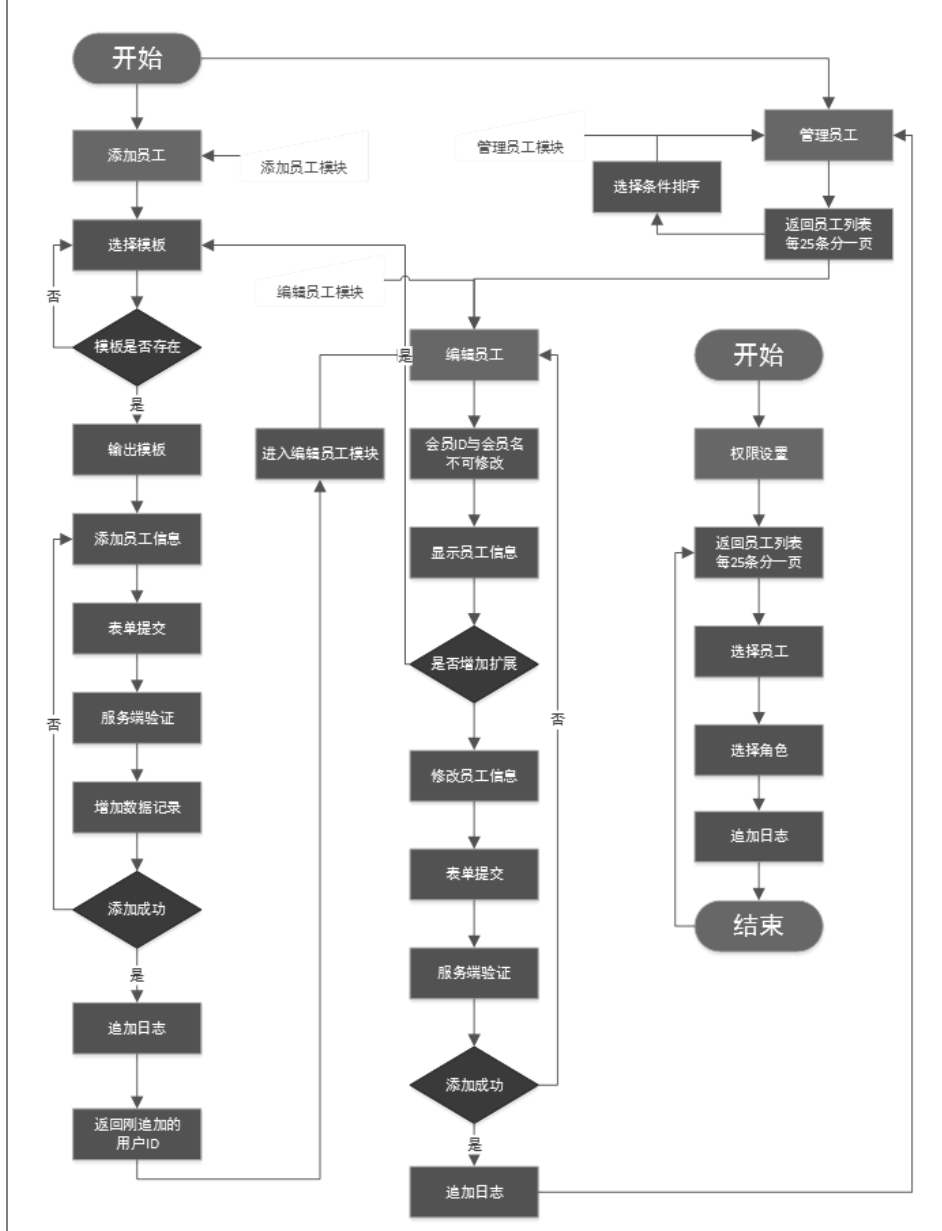
3. 会员管理模块组

会员管理模块组主要包括员工、专家和交易员的管理模块，有添加、查询和编辑等子模块。其中，添加和编辑员工可能需要选择模板，如果不进行选择，则使用默认模板。

在员工管理中，可以使用搜索功能，默认按照添加时间倒序显示，也可以选中其中的某个员工，然后编辑其信息，注意用户 ID 和用户名不能变。

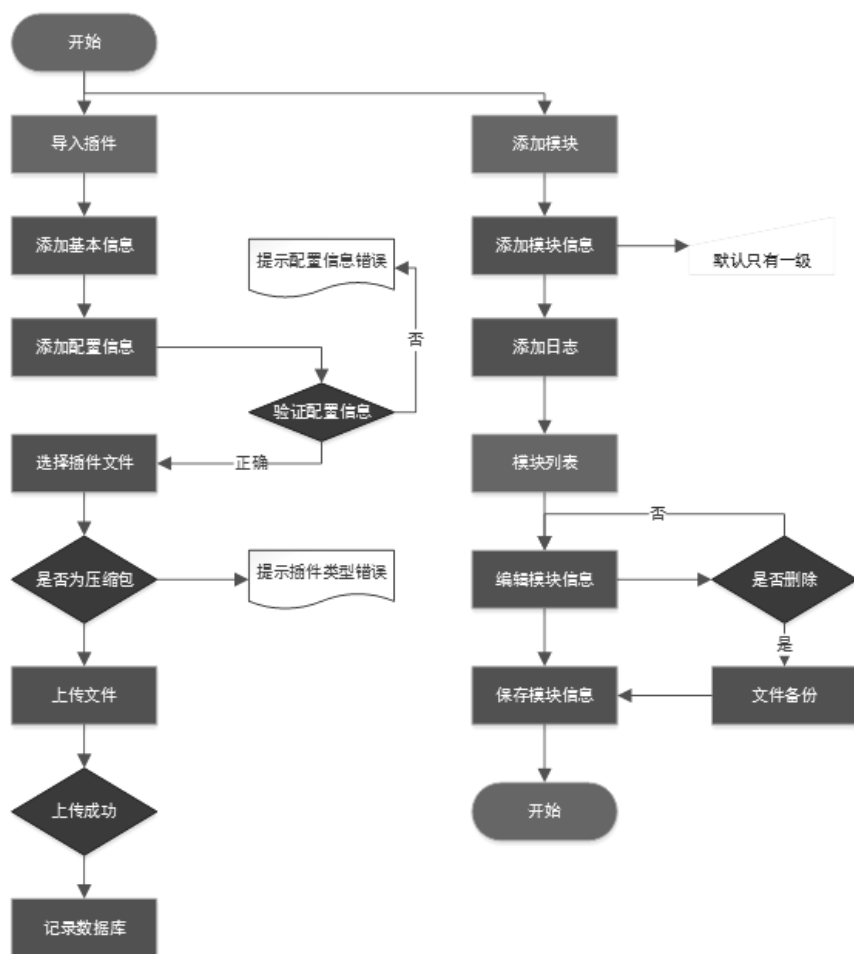
用户的权限设置其实是另一个模块，通过权限设置，按时间倒序显示员工，也可以选中该员工，赋值相应的角色（在数据库中要关联角色表）。

会员管理流程图如下图所示。



4. 模块插件管理模块组

模块插入管理流程图如下图所示。



5. 模板管理

模板管理主要是管理模板表中的 data 字段，data 字段是一个二维数组，这个二维数组实际上就是一个不包含数据的表结构，同时对应这个表结构创建或修改一个数据库的表，类似一个简单的 MySQL 数据库的 Web 客户端。

6. 数据维护

数据维护主要是数据库的连接、导入和导出，技术难度不大。

7. 系统维护

系统维护主要是调用 PHP 的一些目录、文件大小等信息，也可以使用第三方软件。

（二）账户

账户管理的相关内容相对比较简单，一个高级程序员就能完成。

（三）消息

1. 流程模块

动态流程模块以消息的形式通知任务接收人群，任务接收人群可以通过直属上级、直属下级、角色用户和指定员工的方式来设置。任务提示方式可以通过消息和邮件的方式进行通知，默认是消息（也就是站内信）。消息通知人发送的都是只读的内容，无特殊情况，默认是提交者提交的所有信息。接收者可以驳回和通过消息，如果通过，则表示任务完成；如果驳回，则增加驳回记录返回给提交者，任务重新开始。为了方便接收者查看，系统特地生成了一个中间任务表，包括用户提交的数据（JSON 格式）和对应的 URL。

为了方便子流程设置，每个流程都可以选择一个上级流程，但是上级流程必须结束才可以进入下一个子流程，这样流程就形成一个树形的结构了。不过为了简单起见，还是推荐单一线性的流程。

很多时候，用户为了方便，往往是通过职位进行设置的，使得流程模块和组织结构比较紧密地结合在一起，如下图所示为动态流程图。

2. 站内信模块

站内信模块一共有 3 个版本。V1.0 版本仅是一个消息的发送和回复，达到最小可用即可。

V2.0 采用 Socket 编码，增加了历史记录表和表情包，主要满足办公的需求，算是一个公司内部聊天工具。这个聊天工具与我们的工作任务连接紧密，同时也可以自定义一些快速的发起和回复，比如召开部门会议等。

V3.0 是一款比较独立的聊天工具，虽然和 V2.0 使用几乎同样的数据库和服务端，但是它更加独立，是智能办公的一个最小切入点。V3.0 不仅采用了 NLP 技术，对聊天内容进行分析，而且采用了视频对话和远程屏幕控制互动的技术，我们可以将其理解为一个基于办公的智能化聊天工具。

（四）统计分析模块

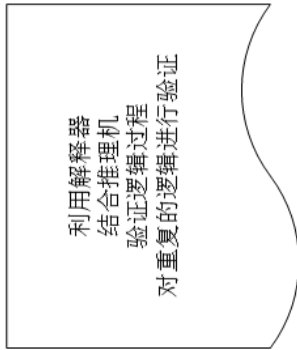
统计分析模块主要是数据库的查询并利用百度的 echarts 进行显示，绝大部分视图在 echarts 官方网站都有示例代码，稍加修改就可以使用，只是自定义视图的工作量比较大。

（五）专家系统之知识库模块

知识库比较简单，主要是一些专家提供的逻辑运算过程，这个逻辑运算过程需要利用解释器（解释器的作用是方便生成底层可运行的代码），然后结合推理机（推理机主要是一个流程图）验证逻辑过程（逻辑过程要结合底层语法），最后为了方便对逻辑过程进行有效的管理，增加用户体验，最好增加比较强大的逻辑验证功能。若没有逻辑验证功能，则要有不错的使用文档，第一期项目时间比较紧，可以适当降低要求。

一般来讲，如果时间允许，至少要开发出一部分可用的事实知识，该事实知识一般都是通过数据库接口实现的，默认为 MySQL 数据库（后期考虑使用 NoSQL 数据库，以提高性能），方便对事实数据库的调用。

同时，为了方便对新事实的追加，可以通过迭代的方式循环追加事实数据库。知识库模块流程图如下图所示。



（六）专家系统之脑电模块——脑电

略。

（七）专家系统之模拟测试模块——模拟

这里比较关键的是要接入一些证券公司的交易接口，由于国内 A 股市场不允许自动化交易，所以在模拟时行情和交易是分开的。开发时要注意一些交易规则，比如不能以涨停价买入、以跌停价卖出等，最好找一个有股票、期货交易经验的程序员来开发。

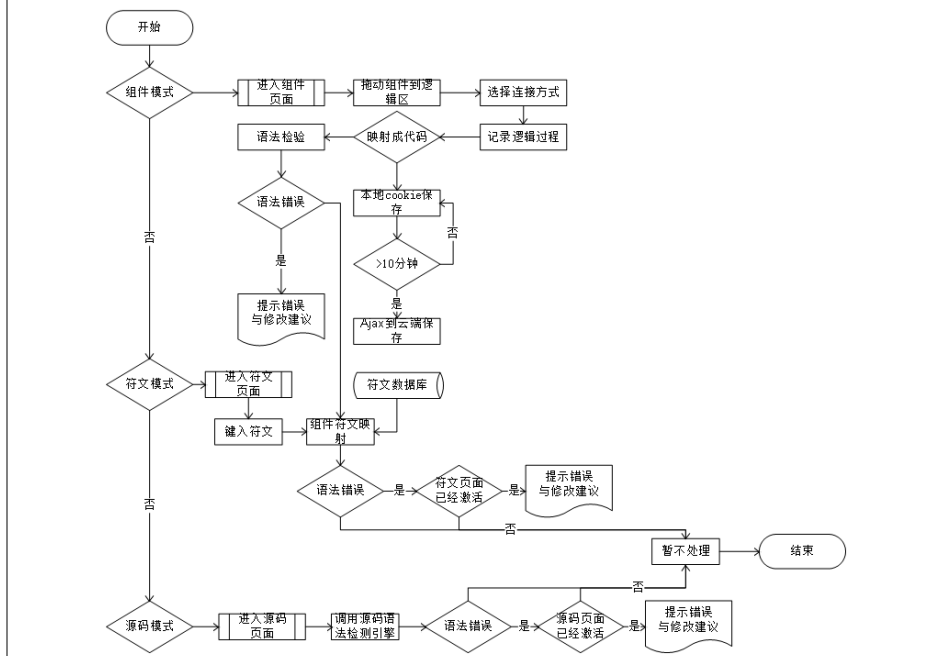
（八）专家系统之推理机模块

推理机模块是一个非常重要的核心模块，为了更好的用户体验，我们采用了可视化编程，这样做的好处是更加灵活。

推理机这个模块交给那些数学好、逻辑清晰，并对似然推理感兴趣的开发小组，该小组由数学顾问亲自领导。

1. 可视化编辑器

推理的 3 种方式是正向推理、逆向推理和双向推理，属于模块的基本架构，这里没有做过多的说明，只是针对用户体验做了一些调整，该调整（其实就是一个简单的可视化编辑器）如下图所示。



2. 决策树与分类树模块

关于决策树与分类树，这里主要是一些算法模型的“增删改查”。

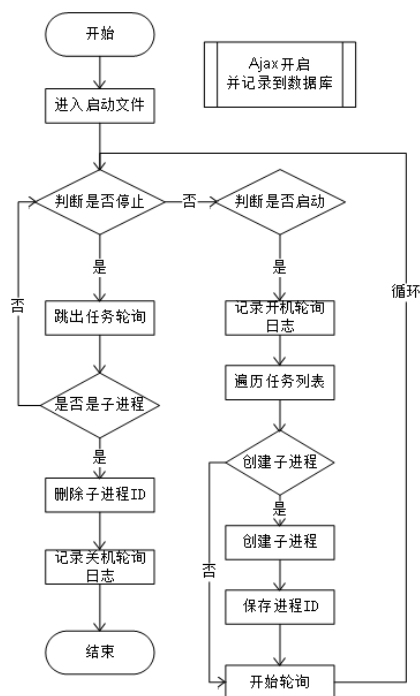
(九) 专家系统之解释器模块

解释器最好能够可视化操作，以图形化显示。这里更加偏重于前端开发，可交由 JavaScript 开发小组完成。

(十) 任务轮询

为了保证系统后台能够不间断地执行任务，我们设置了任务轮询功能。任务轮询是一个循环系统，在没有新参数（如停止、开始、暂停）传入时，系统将不停地遍历所有开启的任务列表。为了简单起见，系统任务主要有图片的生成、任务的处理和日志的处理。

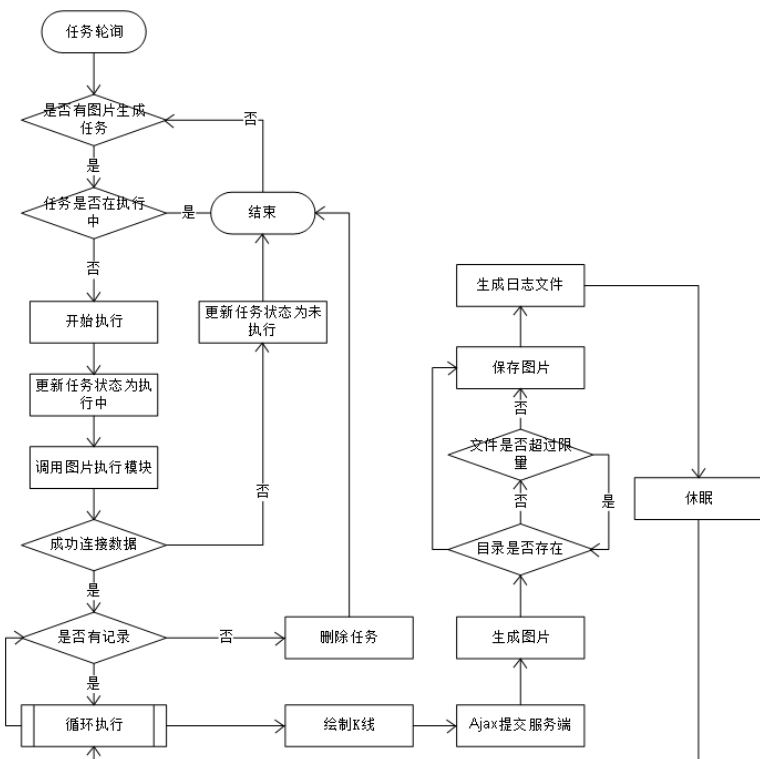
任务轮询流程图如下图所示。



(十一) 图片生成模块

图片生成是数据模块的一个核心功能，主要是为了将已有的 K 线生

成标准化的图片，然后用于机器学习。图片生成的时间可能较长，为了方便管理员实时了解图片生成的进度，我们将图片生成模块分成了三个部分，第一部分是添加生成图片任务的模块，第二部分是修改生成图片任务的模块，第三部分是图片生成中的模块，其中图片生成中的模块有一定的复杂度。图片生成模块如下图所示。



四、设计说明

略。

五、硬件部分

脑电测试是一个独立的开发小组，主要使用 OpenBCI 进行 Arduino 编程开发，然后将检测到的结果输入到数据库中进行机器学习，参考文档有：《原理图》《PCB 图》《BOM 清单》《可编程器件及烧录进制文件》《必要测试点波形图或硬件指标详细说明》《原理详细说明》《与系统内其

他部分接口软硬件详细说明》。

这些参考文档可以以附件的形式单独存放于一个文件夹内。

六、软件的设计结果

（一）自有 PHP 框架

在具体开发中，我们采用自己的一个小型 PHP 框架，框架虽然小，却是一个精简的 MVC 架构，非常方便系统的开发、测试和维护。这个框架统一采用 URL 传参的方式进行调用，默认用 index.php 作为框架的入口文件。

框架源码部分略。

（二）PHP 机器学习框架

PHP-ML 是一款不错的机器学习框架，为了方便进行机器学习，在项目测试阶段主要使用该框架。

参考网址是 <https://www.oschina.net/p/php-ml/>。

（三）JavaScript 框架

1. echarts.js

echarts.js 是百度公司提供的一款商业级数据图表统计软件，功能强大，也符合我国用户的使用习惯，适合进行基本的图表统计分析。

2. jquery.min.js

JQuery 是一个快速、简捷的 JavaScript 框架，里面融合了很多功能，特别是在元素的选择和控制上非常灵活。很多常用的功能用一两个函数就可以轻松搞定，这里选用 jquery.min.js 作为 JavaScript 的基本框架，以提高开发效率。

2.7 数据库设计

数据是系统的重要组成部分，好比汽车中的汽油，是系统正常运行的

保障，尤其是 B/S 架构的互联网项目。数据库设计之所以单独拿出来，而不是交给程序员设计，主要是为了项目安全和并行开发考虑的。我们不鼓励程序员直接拥有修改数据字段的权限，这就要求数据库设计师对产品的需求和模块设计有深入的了解，如果对产品理解不够深入，可能就会出现程序员要求不断修改数据库字段的情况，比如增加字段或修改字段类型与长度。

数据库设计说明书的范文如下。

量化投资专家系统 数据库设计说明书

拟 制 人：王昭东

审 核 人：副总经理

批 准 人：总经理

2017 年 12 月 30 日

一、引言

（一）编写目的

（1）方便技术开发人员熟悉数据库结构，提高开发速度，促使软件开发与数据分离，方便升级和维护，有利于数据安全。

（2）制定一些数据库的设计规范，方便数据的日后扩展、升级和维护。

（二）背景

（1）专家系统目前主要采用 4 个数据库，分别是后台管理数据库、推理机知识库、会员数据库和消息库。另外，还有统计分析数据库、插件数据库、客户关系管理数据库和日志数据库等。

（2）数据库设计负责人是 DBA 和系统架构师，其中 DBA 负责具体实现，系统架构师负责设计和检验。

(三) 定义

无。

(四) 参考资料

无。

二、外部设计

(一) 标识符的状态

关系型数据库：MySQL5.5+、UTF-8 编码。

(1) 后台管理数据库，数据库名称为 ai_admin，用于系统后台管理。

(2) 推理机知识库，数据库名称为 ai_tree，用于推理机逻辑树的管理。

(3) 统计分析数据库，数据库名称为 ai_sql，用于关系型数据库的统计分析。

(4) 会员管理数据库，数据库名称为 ai_member，用于会员与员工管理。

(5) 推理机知识库数据库，数据库名称为 ai_tree，用于管理策略。

(6) 客户关系数据库，数据库名称为 ai_crm，用于客户关系管理。

非关系型数据库：MongoDB，UTF-8 编码。

(1) 消息数据库，数据库名称为 ai_im，用于记录聊天记录。

(2) 日志数据库，数据库名称为 ai_log，用于记录系统日志。

(二) 使用它的程序

(1) 专家系统采用 PHP 进行 MySQL 数据库的访问，版本和数据库一一对应，严格区分测试版和正式版（V1.0 还不区分测试版与正式版，V2.0 以后开始有此区别）。

(2) 机器学习框架，用 PHP 访问 MySQL 数据库。

(三) 约定

无，采用敏捷开发，注意事后日志的记录，要求团队配合密切，要求有相关开发经验的人员来完成，尽可能保证项目模块的开发进度。

(四) 支持软件

PHP7、phpMyAdmin、Python3.5、Navicat for MySQL

三、结构设计

(一) 概念结构设计

数据库默认为 MySQL 和 NoSQL。

1. 数据库前缀

如 ai_dataname、ai_admin (后台管理数据库)。

2. 表前缀

如 dataname_tablename、admin_logo (后台管理日志表)。

3. 字段名称

英文小写字母且无下画线，如 userid (用户 ID)。

4. 数据类型

字节少的优先。

5. 数据长度

数据长度够用就行。

6. 分区与分表

可以根据实际情况划分。

7. 关于索引

原则上不超过两个。

四、运用设计

(一) 数据字典设计

1. 后台管理数据库 ai_admin

使用 UTF-8 编码。

管理员用户表 (admin) 如下表所示。

字段	类型	长度	默认	注释	主键
userid	int	10	0	用户 id	*自增长, 主键
username	varchar	20	Null	用户登录名	*
password	varchar	32	Null	用户密码	*
roleid	smallint	5	0	角色 id	*
encrypt	varchar	6	Null	加密因子	*
lastloginip	varchar	15	Null	最后登录 IP	*
lastlogintime	int	10	0	最后登录时间	*
email	varchar	40	Null	用户邮箱	*
phone	varchar	13	Null	用户手机	
realname	varchar	50	Null	用户真实姓名	
card	varchar	255	Null	密保卡	
locktime	int	10	0	锁定时间	
error	tinyint	2	0	错误次数	
disabled	tinyint	1	0	是否禁用	

管理员用户权限表 (admin_role) 如下表所示。

字段	类型	长度	默认	注释	主键
roleid	tinyint	3	0	角色 id	*自增长, 主键
rolename	varchar	50	Null	角色名称	*
description	varchar	50	Null	角色说明	
adduserid	int	10	0	添加角色用户 id	*
disabled	tinyint	1	0	是否禁用	*
roles	varchar	20	Null	子角色集合, 注意死循环	
modules	varchar	500	Null	模块权限集合	*

模块表 (admin_module) 如下表所示。

字段	类型	长度	默认	注释	主键
moduleid	mediumint	5	0	模块 id	*自增长, 主键
pid	mediumint	5	0	父模块 id	
name	varchar	20	Null	作者名称	
url	varchar	75	Null	模块路径	*
version	varchar	50	Null	模块版本	
description	varchar	255	Null	模块描述	
functions	varchar	255	Null	函数集合	
Listorder	tinyint	3	0	模块排序	*
disabled	tinyint	1	0	是否禁用	*
installdate	int	10	0	安装时间	*

续表

字段	类型	长度	默认	注释	主键
module	varchar	20	Null	模块名称	
thumbnail	varchar	50	Null	缩略图	
token	varchar	32	Null	Token（令牌）	
type	tinyint	1	0	用户类型，0 为会员，1 为管理员	
show	tinyint	1	0	是否显示到菜单中，0 为显示，1 为不显示	
disabled	tinyint	1	0	是否禁用，0 为启用，1 为禁用	

文件插件表（admin_files）如下表所示。

字段	类型	长度	默认	注释	主键
fileid	int	10	0	文件插件 id	*自增长，主键
pid	int	10	0	父插件 id	*
filename	char	50	Null	文件插件名称	
filepath	char	200	Null	文件插件路径	
filesize	int	10	0	文件插件大小	*
fileext	char	10	Null	文件插件扩展名	
filetype	varchar	50	Null	文件插件类型	
userid	mediumint	8	0	上传者用户 id	
username	varchar	20	Null	上传者用户名	*
uploadtime	int	10	0	上传时间	*
uploadip	char	15	Null	上传者 IP 地址	*
dataset	varchar	250	Null	数据库设置	*
pageset	varchar	250	Null	页面设置	*
listorder	tinyint	3	0	文件插件排序	*
thumbnail	varchar	50	Null	文件插件缩略图	
disabled	tinyint	1	0	是否禁用	

数据接口表（admin_interface）如下表所示。

字段	类型	长度	默认	注释	主键
interfaceid	int	10	0	接口 id	*自增长，主键
name	varchar	50	0	接口名称	*
url	varchar	50	Null	接口地址	*
interfaceuser	Varchar	50	Null	接口账户	*
interfacepassword	varchar	50	0	接口密码	*
userid	mediumint	6	Null	添加者 id	*
token	varchar	32	Null	Token（令牌）	
price	int	10	0	价格，单位为分	

续表

字段	类型	长度	默认	注释	主键
startdate	datetime	0	—	开始时间	
enddate	datetime	0	—	结束时间	
msgdate	datetime	0	—	提醒时间	
contacts	varchar	50	Null	联系方式	
codefile	varchar	150	Null	示例文档	
helpfile	varchar	150	0	帮助文档	

管理员通知表 (admin_announce) 如下表所示。

字段	类型	长度	默认	注释	主键
aid	smallint	4	0	通知 id	*自增长, 主键
type	tinyint	1	0	通知类型, 0 为系统, 1 为管理员, 2 为会员	*member_menu_shortcut
title	char	80	0	标题	*
url	varchar	150	Null	通知路径	
starttime	int	10	Null	开始时间	
endtime	int	10	0	逾期时间	
fromid	int	10	Null	请求者会员 id	
userid	int	10	Null	接收者, 管理员 id	
addtime	int	10	0	添加时间	
disabled	tinyint	1	0	是否完成, 0 为未完成, 1 为完成	
adminname	varchar	25	Null	管理员用户名	
updatetime	int	10	0	完成时间	

管理员快捷菜单表 (member_menu_shortcut) 如下表所示。

字段	类型	长度	默认	注释	主键
userid	mediumint	5	0	管理员 id	*admin, 主键
roles	varchar	200	Null	权限集合	*

管理员后台自定义菜单表 (admin_menu) 如下表所示。

字段	类型	长度	默认	注释	主键
menuid	mediumint	5	0	会员菜单 id	*自增长, 主键
menuname	varchar	30	Null	菜单名称	*
description	varchar	30	Null	菜单描述	*
listorder	smallint	5	0	菜单排序	*
disabled	tinyint	1	0	是否禁用	*

续表

字段	类型	长度	默认	注释	主键
userid	mediumint	6	0	会员 id	*
roles	varchar	200	Null	权限集合	*
modules	varchar	200	Null	模块集合	
ico	varchar	150	Null	菜单图标	
pid	mediumint	5	0	上级菜单 id	

2. 会员管理数据库 ai_member

使用 UTF-8 编码。

会员表 (member) 如下表所示。

字段	类型	长度	默认	注释	主键
memberid	mediumint	8	0	会员 id	*自增长, 主键
membername	char	20	Null	会员名	*
password	char	32	Null	会员密码	*
encrypt	char	6	Null	加密因子	*
nickname	char	20	Null	会员昵称	
regdate	int	10	0	注册日期	*
lasttime	int	10	0	最后登录时间	*
regip	char	15	Null	注册 IP	
lastip	char	15	Null	最后登录 IP	
error	smallint	5	0	错误次数	
email	char	32	Null	会员邮箱	
roleid	tinyint	3	0	会员权限 id	
point	smallint	5	0	会员积分	
message	tinyint	1	0	会员消息, 0 为无	
vip	tinyint	1	0	VIP 会员级别	
phone	Char	11	Null	会员电话	
disabled	tinyint	1	0	是否禁用	*
image	varchar	150	Null	会员头像	
outdate	int	10	0	会员离职时间	

会员默认信息表 (member_default) 如下表所示。

字段	类型	长度	默认	注释	主键
id	int	11	0	会员基本信息 id	*自增长, 主键
userid	int	11	0	会员 id	*
photo	varchar	100	Null	会员照片	
fullname	char	20	Null	会员真实姓名	

续表

字段	类型	长度	默认	注释	主键
hisusername	char	30	Null	亲友会员名称	
hisfullname	char	30	Null	亲友真实名称	
kinsfolk	char	10	Null	亲情关系	
education	char	10	Null	学历	
address	varchar	50	Null	常驻地址	
area	char	6	0	区号	
tel	char	15	0	电话	
qq	char	15	0	QQ	
birthday	int	11	0	生日	
marriage	int	11	0	结婚纪念日	
sex	tinyint	1	0	性别	
email	char	32	Null	常用邮箱	
china	char	15	Null	省份	
city	char	20	Null	城市	
blood	char	5	Null	血型	
mobile	char	11	0	手机	*
text	varchar	1000	Null	简介	

会员工作信息表 (member_job) 如下表所示。

字段	类型	长度	默认	注释	主键
id	int	11	0	会员工作简历 id	*自增长, 主键
userid	int	11	0	会员 id	*
bname	varchar	30	Null	单位名称	
address	varchar	50	Null	单位地址	
area	char	6	0	单位区号	
tel	char	15	0	单位电话	
qq	char	15	0	证明人 QQ	
addtime	int	11	0	入职时间	
outtime	int	11	0	离职时间	
work	char	30	Null	职务职位	
email	char	32	Null	证明人邮箱	
china	char	15	Null	单位所在省份	
city	char	20	Null	单位所在城市	
mobile	char	11	Null	证明人手机	
text	varchar	1000	Null	工作简介	
image	varchar	50	Null	离职证明扫描件	

专家会员表 (expert_data) 如下表所示。

字段	类型	长度	默认	注释	主键
memberid	mediumint	8	0	会员 id	*主键
level	tinyint	1	0	专家级别	*
newgame	varchar	255	Null	最新策略	

会员积分往来表 (member_points) 如下表所示。

字段	类型	长度	默认	注释	主键
id	int	10	0	积分往来 id	*自增长, 主键
frommembername	varchar	25	0	发送者的用户名	*
tomembername	varchar	25	0	接收者的用户名	*
type	tinyint	2	0	积分类型	*
tod	tinyint	1	0	是否是自己购买, 1 为自己购买, 0 为他人购买	*
time	int	10	0	发生时间	*
number	int	11	0	积分数	*
status	tinyint	1	0	状态表, 1 表示已接收, 0 表示等待确认	*
text	varchar	250	Null	说明	

会员自定义菜单表 (member_menu) 如下表所示。

字段	类型	长度	默认	注释	主键
menuid	mediumint	5	0	会员菜单 id	*自增长, 主键
menuname	varchar	30	Null	菜单名称	*
description	varchar	30	Null	菜单描述	*
listorder	smallint	5	0	菜单排序	*
disabled	tinyint	1	0	是否禁用	*
userid	mediumint	6	0	会员 id	*
roles	varchar	200	Null	权限集合	*
modules	varchar	200	Null	模块集合	
ico	varchar	150	Null	菜单图标	
pid	mediumint	5	0	上级菜单 id	

会员快捷菜单表 (member_menu_shortcut) 如下表所示。

字段	类型	长度	默认	注释	主键
userid	mediumint	5	0	会员 id	*member, 主键
roles	varchar	200	Null	权限集合	*

商品信息表 (goods) 如下表所示。

字段	类型	长度	默认	注释	主键
id	int	10	0	商品 id	*自增长, 主键
type	tinyint	1	0	0 为混合策略, 1 为 CTA 策略, 2 为高频策略, 3 为突破策略	
name	varchar	50	Null	产品名称	
manager	varchar	20	Null	基金经理	
target	varchar	30	Null	投资方向	
sum	int	10	Null	产品总值	
day	int	10	Null	产品周期 (天)	
old	int	10	Null	已经认购	
mini	int	10	Null	最低认购	
bank	varchar	20	0	开户银行	
enddate	date	0	0	结束日期	
yield	varchar	20	0	预期收益	
isid	varchar	20	0	备案编号	
upfile	varchar	100	0	附件地址	
description	varchar	1000	0	产品简介	
hot	int	10	Null	热度排序	
disabled	tinyint	1	Null	0 为募集中, 为 1 已关闭, 2 为已撤销	
userid	int	10	Null	会员 id	
addtime	int	10	Null	添加时间	

财务凭证表 (finance_accvouch) 如下表所示。

字段	类型	长度	默认	注释	主键
id	int	10	0	商品 id	*自增长, 主键
type	tinyint	1	0	0 为混合策略, 1 为 CTA 策略, 2 为高频策略, 3 为突破策略	
name	varchar	50	Null	产品名称	
manager	varchar	20	Null	基金经理	
target	varchar	30	Null	投资方向	
sum	int	10	Null	产品总值	
day	int	10	Null	产品周期 (天)	
old	int	10	Null	已经认购	
mini	int	10	Null	最低认购	
bank	varchar	20	0	开户银行	
enddate	date	0	0	结束日期	
yield	varchar	20	0	预期收益	
isid	varchar	20	0	备案编号	

续表

字段	类型	长度	默认	注释	主键
upfile	varchar	100	0	附件地址	
description	varchar	1000	0	产品简介	
hot	int	10	Null	热度排序	
disabled	tinyint	1	Null	0 为募集中, 1 为已关闭, 2 为已撤销	
userid	int	10	Null	会员 id	
addtime	int	10	Null	添加时间	

3. 日志管理数据库 ai_log

使用 UTF-8 编码, 暂时用 MySQL 数据库, 以后再进行迁移。

日志管理表 (log) 按照时间分表, 默认每周一次, 表格式为 log_20160107, 如下表所示。

字段	类型	长度	默认	注释	主键
logid	int	10	0	日志 id	*自增长, 主键
type	int	10	0	日志类型, 1 为导入, 2 为导出, 3 为备份, 4 为错误, 5 为警告	*
querystring	varchar	255	Null	请求内容	
data	mediumtext	0	Null	返回内容	
userid	mediumint	8	0	操作者 id	*
username	varchar	20	Null	操作者的用户名	
ip	varchar	15	Null	IP 地址	
time	datetime	0	0	操作时间	*
isread	tinyint	1	0	是否阅读, 0 为未读, 1 为已读, 2 为删除	*

会员日志管理表 (member_log) 按照时间分表, 默认每周一次, 表格式为 member_log_20160107, 如下表所示。

字段	类型	长度	默认	注释	主键
logid	int	10	0	日志 id	*自增长, 主键
type	int	10	0	日志类型, 1 为导入, 2 为导出, 3 为备份, 4 为错误, 5 为警告	*
querystring	varchar	255	Null	请求内容	
data	mediumtext	0	Null	返回内容	

续表

字段	类型	长度	默认	注释	主键
memberid	mediumint	8	0	会员操作者 id	*
membername	varchar	20	Null	会员操作者的用户名	
ip	varchar	15	Null	IP 地址	
time	datetime	0	0	操作时间	*
israde	tinyint	1	0	是否阅读, 0 为未读, 1 为已读, 2 为删除	*

专家系统日志管理表 (tree_log) 同 member_log 表。

后台日志管理表 (admin_log) 按照时间分表, 默认每周一次, 表格式为 admin_log_20160107, 如下表所示。

字段	类型	长度	默认	注释	主键
logid	int	10	0	日志 id	*自增长, 主键
type	int	10	0	日志类型, 1 为导入, 2 为导出, 3 为备份, 4 为错误, 5 为警告	*
title	varchar	25	null	操作内容	
querystring	varchar	255	Null	请求内容	
data	mediumtext	0	Null	返回内容	
adminid	mediumint	8	0	会员操作者 id	*
adminname	varchar	20	Null	会员操作者用户名	
ip	varchar	15	Null	IP 地址	
time	datetime	0	0	操作时间	*
israde	tinyint	1	0	是否阅读, 0 为未读, 1 为已读, 2 为删除	*

图片生成日志表 (img_log) 如下表所示。

字段	类型	长度	默认	注释	主键
logid	int	10	0	日志 id	*自增长, 主键
username	varchar	30	0	用户名称	*
md5	varchar	32	null	图片 MD5	*
path	varchar	255	null	图片路径	*
time	datetime	0	Null	生成时间	*

4. 消息通知管理数据库 ai_im

使用 UTF-8 编码。

通知表 (member_announce) 如下表所示。

字段	类型	长度	默认	注释	主键
aid	smallint	4	0	通知 id	*自增长, 主键
type	tinyint	1	0	任务类型, 0 为公告, 1 为系统, 2 为会员	*
title	char	80	0	标题	*
url	varchar	150	Null	通知路径	
starttime	int	10	Null	开始时间	
endtime	int	10	0	逾期时间	
fromid	int	10	Null	请求者会员 id	
memberid	int	10	Null	接收者会员 id	
addtime	int	10	0	添加时间	

备忘录表 (member_memorandum) 如下表所示。

字段	类型	长度	默认	注释	主键
id	int	10	0	备忘录 id	*自增长, 主键
type	tinyint	1	0	任务类型, 0 为一般, 1 为紧急, 2 为非常紧急	
title	char	50	0	标题	
text	varchar	250	Null	内容	
starttime	date	10	Null	开始日期	
endtime	date	10	0	逾期日期	
usertype	tinyint	1	0	会员类型	
memberid	int	10	Null	接收者会员 id	
addtime	int	10	0	添加时间	

消息表 (message) 如下表所示。

字段	类型	长度	默认	注释	主键
msmid	int	10	0	消息 id	*自增长, 主键
fromname	varchar	35	0	发送者会员	*
toname	varchar	35	0	接收者会员	
status	tinyint	1	0	消息状态, 0 为未读, 1 为已读, 2 为草稿箱	
time	int	10	0	发送时间	
subject	char	80	Null	标题	
content	varchar	500	0	消息内容	
replyid	int	10	0	上级消息 id	
disabled	tinyint	1	0	是否删除, 0 为正常, 1 为删除	

消息扩展表 (message_data) 如下表所示。

字段	类型	长度	默认	注释	主键
msmid	int	10	0	消息 id	* message 主键
content	mediumtext	0		消息内容	*
view	varchar	100	Null	显示方式	
template	varchar	30	Null	显示模板	
file	varchar	150	Null	附件路径	

消息设置表 (message_set) 如下表所示。

字段	类型	长度	默认	注释	主键
memberid	int	10	0	会员 id	* member 表主键
emailset	varchar	500	0	邮件设置	
autograph	varchar	500	0	签名设置	

组织机构列表 (member_group) 如下表所示。

字段	类型	长度	默认	注释	主键
groupid	int	10	0	职位 id	*自增长, 主键
groupname	varchar	35	0	职位名称	*
mainusername	varchar	50	0	负责人姓名	
time	int	10	0	添加时间	
text	char	80	0	岗位简介及职责	
members	varchar	500	0	下属会员集合, 分隔	
pids	varchar	35	0	上级 id 集合, 分隔	
disabled	tinyint	1	0	是否删除, 0 为正常, 1 为禁用	

会员工作流程表 (member_flow) 如下表所示。

字段	类型	长度	默认	注释	主键
flowid	int	10	0	流程 id	*自增长, 主键
frompid	int	10	0	父流程 id	
pid	tinyint	1	0	上级中转流程 id	
name	char	50	Null	流程名称	*
fromtype	tinyint	1	0	发起人类型, 0 为指定员工, 1 为指定角色, 2 为指定模块	*
fromname	varchar	20	Null	发起人会员名	
fromroleid	tinyint	2	0	发起者角色 id	
frommoduleid	int	10	0	发起者模块 id	
sendtype	tinyint	1	0	通知方式 0 为通知, 1 为邮件, 2 为消息, 3 为邮件和消息	*

续表

字段	类型	长度	默认	注释	主键
sendclick	tinyint	1	0	触发条件, 0 为增加一条记录, 一般以 disabled 字段为 0 作为判断依据	*
totype	tinyint	1	2	接收人类型, 0 为指定员工, 1 为指定角色, 2 为直属上级, 3 为直属下级, 4 为指定模块	*
toname	varchar	20	0	接收人用户名	
toroleid	tinyint	2	0	接收人角色 id	
tomoduleid	int	10	0	接收人模块 id	
agree	tinyint	3	0	通过率, 最大为 100	*
outdate	tinyint	2	0	最大逾期时间, 单位为天	
addtime	int	10	0	流程添加时间	
addusername	varchar	25	0	添加者用户名	
isend	tinyint	1	0	流程是否结束, 0 为没有结束, 1 为结束	
isgood	tinyint	1	0	流程是否正常, 0 为正常, 1 为无此角色, 2 为无此用户, 3 为无直属上级, 4 为无直属下级	*
disabled	tinyint	1	0	是否禁用, 0 为启用, 1 为禁用	*
iswork	tinyint	1	0	是否轮询, 0 为未轮询, 1 为轮询中	

流程日志表 (flow_log) 如下表所示。

字段	类型	长度	默认	注释	主键
aid	smallint	4	0	通知 id	*自增长, 主键
agree	tinyint	1	1	是否通过, 0 为通过, 1 为不通过	*
title	char	80	0	标题	*
url	varchar	150	Null	通知路径	
starttime	int	10	Null	开始时间	
endtime	int	10	0	逾期时间	
fromid	int	10	Null	请求者会员 id	
memberid	int	10	Null	接收者会员 id	
addtime	int	10	0	添加时间	

5. 推理机策略管理数据库 ai_tree

使用 UTF-8 编码。

策略表 (games) 如下表所示。

字段	类型	长度	默认	注释	主键
gameid	mediumint	6	0	策略 id	*自增长, 主键
username	varchar	20	Null	申请人账户名	*
refereename	varchar	250	Null	推荐人账户名	
name	varchar	35	Null	策略名称	
realweight	Int	5		真实权重	
testweight	int	5	0	测试权重	
mockweight	int	5	0	模拟权重	
expertweight	int	5	0	专家权重	
exchange	tinyint	1	0	市场类型, 0 为 A 股, 1 为港股, 2 为郑商, 3 为大商, 4 为上期, 5 为中金所, 6 为 A 股创业板	
goods	varchar	350	Null	品种合约, 股票代码	
yearyield	char	10	Null	年化收益率	
sharp	char	10	Null	夏普率	
loopback	char	10	Null	最大回测	
description	varchar	50	Null	策略思想	
codefile	varchar	150	Null	策略源码	
testing	tinyint	1	0	测试类型, 0 为测试, 1 为模拟	
startdate	int	10	0	开始时间	
enddate	int	10	0	结束时间	
disabled	tinyint	1	0	是否使用, 0 为启用, 1 为禁用	

咒语字典和解释器表 (incantation) 如下表所示。

字段	类型	长度	默认	注释	主键
lanid	mediumint	6	0	语言字典 id	*自增长, 主键
incantation	varchar	250	Null	咒语符文字符	*唯一主键
type	tinyint	1	0	符文类型, 0 为语法结构, 1 为函数, 2 为字符串, 3 为数组, 4 为数据库, 5 为日期, 6 为文件, 7 为网络	
php	varchar	250	Null	PHP 字符	*
description	varchar	150	Null	语法说明	
case	varchar	1000	Null	示例	
error	varchar	250	Null	错误, 说明	
codefile	varchar	150	Null	demo 源码	
disabled	tinyint	1	0	是否禁用	*

生成图片表（image_work）如下表所示。

字段	类型	长度	默认	注释	主键
id	mediumint	8	0	语言字典 id	*自增长, 主键
username	varchar	30	Null	会员名称	
type	varchar	10	Null	图片类型	
dataid	int	10	0	数据库接口 id	
k	varchar	80	Null	K 线类型	
fun	varchar	50	Null	被调用的函数名称	
up	int	10	0	上涨时间跨度	1
down	int	10	0	下跌时间跨度	1
unit	varchar	80	Null	K 线周期类型	
width	int	10	0	宽度, 单位是像素	
height	int	10	0	高度, 单位是像素	
name	char	20	Null	命名规则	
maxfile	int	10	0	最大文件数	
path	varchar	50	Null	存储路径	
sql	varchar	200	Null	SQL 语句	
slep	int	5	0	轮询时间, 单位为秒	
email	varchar	50	Null	邮箱	
woking	tinyint	1	0	工作状态, 0 为停止工作, 1 为工作中, 2 为暂停	
disabled	tinyint	1	0	是否禁用, 0 为启用, 1 为禁用	*

推理机列表（tree）如下表所示。

字段	类型	长度	默认	注释	主键
treeid	mediumint	8	0	决策树 id	*自增长, 主键
type	varchar	10	Null	决策树类型	*
title	varchar	80	Null	决策树标题	
description	mediumtext	0	Null	决策树描述	
listorder	tinyint	3	0	排序	
username	char	20	Null	作者会员名称	*
inputtime	int	10	Null	添加时间	
updatetime	int	10	0	更新时间	
disabled	tinyint	1	0	是否禁用, 0 为启用, 1 为禁用	*

推理机扩展表 (tree_data) 如下表所示。

字段	类型	长度	默认	注释	主键
treeid	mediumint	8	0	决策树 id	*tree 表主键
content	mediumtext	0	0	决策树表达式	
template	varchar	30	Null	决策树模板	

事实库列表 (fact) 如下表所示。

字段	类型	长度	默认	注释	主键
factid	mediumint	8	0	知识库事实 id	*自增长, 主键
reg	varchar	10	Null	匹配正则	*
pids	varchar	80	Null	多个事实父 id, 默认一个	
description	varchar	100	Null	事实特征描述	
status	tinyint	1	0	是否是事实, 0 为不是, 1 为是	
username	char	20	Null	添加者会员名称	*
time	int	10	0	添加时间	
ratio	int	5	0	概率百分比, 万分之一, 默认为 0	
disabled	tinyint	1	0	是否禁用, 0 为启用, 1 为禁用	*

文章列表 (blog) 如下表所示。

字段	类型	长度	默认	注释	主键
blogid	mediumint	8	0	文章 id	*自增长, 主键
type	varchar	10	Null	文章类型, 0 为评价标准	*
title	varchar	80	Null	文章标题	
description	mediumtext	0	Null	文章描述	
listorder	tinyint	3	0	排序	
username	char	20	Null	作者会员名称	*
inputtime	int	10	0	添加时间	
updatetime	int	10	0	更新时间	
disabled	tinyint	1	0	是否禁用, 0 为启用, 1 为禁用	*

文章扩展表 (blgo_data) 如下表所示。

字段	类型	长度	默认	注释	主键
blogid	mediumint	8	0	文章 id	*blog 表主键
content	mediumtext	0	0	文章内容	
template	varchar	30	Null	显示模板	

常用函数列表 (function) 如下表所示。

字段	类型	长度	默认	注释	主键
funid	mediumint	6	0	函数 id	*自增长, 主键
style	tinyint	2	0	样式, 0 为指标, 1 为财务, 2 为行情, 3 为交易指令, 4 为系统命令, 5 为数学	
type	tinyint	1	Null	函数类型, 0 为 SQL 函数, 1 为控件, 2 为插件, 3 为接口	*
funname	varchar	250	Null	函数名称	
usertype	tinyint	1	1	用户类型, 0 为管理员, 1 为会员	
username	varchar	35	Null	用户名称	
php	varchar	250	Null	PHP 底层字符	
description	varchar	35	0	语法说明	
case	varchar	1000	Null	示例	
error	varchar	250	0	错误与警告	
sql	varchar	500	Null	SQL 语句	
hot	int	10	0	使用次数	
interface	varchar	500	Null	接口数组	
show	tinyint	1	0	是否显示到菜单中, 0 为显示, 1 为不显示	*
md5	char	32	Null	MD5 验证, 针对实际命令验证	
codefile	varchar	150	Null	demo 源码	
disabled	tinyint	1	0	是否禁用, 0 为启用, 1 为禁用	*

推理机临时存储表 (expert) 如下表所示。

字段	类型	长度	默认	注释	主键
treeid	int	10	0	临时推理机 id	*自增长, 主键
title	varchar	20	null	临时推理机标题	*
content	mediumtext	0	0	推理机表达式	
type	tinyint	1	0	编辑区类型, 0 为组件, 1 为符 文, 2 为源码, 3 为 SQL 语句	*
time	int	10	0	保存时间	
membername	varchar	30	0	作者名称	*

6. 统计分析数据库 ai_count

使用 UTF-8 编码。

用户自定义数据接口表 (data_set) 如下表所示。

字段	类型	长度	默认	注释	主键
dataid	int	10	0	数据接口 id	*自增长, 主键
title	varchar	30	null	数据名称	
set	varchar	1000	null	接口内容数组	
usertype	tinyint	1	0	用户类型, 0 为后台, 1 为前台	*
username	varchar	35	null	会员名称	*
type	varchar	20	null	数据类型	
time	int	10	0	添加时间	
disabled	tinyint	1	0	是否禁用, 0 为启用, 1 为禁用	*

7. 客户管理数据库 ai_crm

投资人表 (investor) 如下表所示。

字段	类型	长度	默认	注释	主键
userid	int	10	0	投资人 id	*自增长, 主键
type	tinyint	1	1	0 为未分配, 1 为私有, 2 为共享	
username	char	20	Null	账户名	
password	char	32	Null	密码	
encrypt	char	6	Null	加密因子	
nickname	char	20	Null	投资人姓名	
regdate	int	10	Null	注册日期	
lasttime	int	10	0	最后登录时间	
regip	char	15	Null	注册 IP	
lastip	char	15	Null	最后登录 IP	
error	smallint	5	0	错误次数	
email	char	32	Null	会员邮箱	
point	smallint	5	Null	会员积分	
message	tinyint	1	0	会员消息, 0 为无	
vip	tinyint	1	0	VIP 会员级别	
phone	char	11	Null	会员电话	
image	varchar	150	Null	会员头像	
nominate	char	20	Null	推荐人	
disabled	tinyint	1	0	是否禁用, 0 为启用, 1 为禁用	
adduser	varchar	30	Null	添加者姓名	

(二) 安全保密设计

由于是内部使用, 所以更强调物理隔离。另外, 由于项目采用迭代开发, 所以第一期的安全性能等并不是本书重点。

2.8 网页设计

当一个需求说明文档或产品原型图做出来后，就可以进行具体的网页设计工作了。当然，这里有一个前提，就是产品原型图最好让技术总监过目，否则实现不了就不好了。网页设计和产品原型一般属于产品总监的管辖范围。

若是小公司，还可以把网页设计这个项目外包出去，也就是说产品部只负责市场调研和产品原型即可。

如果有合作伙伴，交待产品原型图并大概说一下基本要求就可以了，尤其是定制开发的管理软件要求一般不多，设计要求的内容大致如下。

(1) 项目：量化投资专家系统（金融行业管理软件）。

(2) 特点：科幻、智能、简捷、布局合理。

(3) 主色调：蓝色、青色、白色。

(4) 主要界面（比较有代表性的页面）：登录页面、列表页面（左侧菜单、顶部菜单、列表）、发送页面（基本表单）和常用控件页面（其他按钮、文本框、浮动窗口等）。

能够提供几个设计参考网站那是最好的，参考网站不用太多，1~3 个就好。主要目的就是让设计师明白，我们在做一个什么东西，以及现在市面上有哪些竞品。

设计师一般会给几个效果图，当选中其中一个后，设计师就会设计具体的页面并切成图（有些大型公司，设计和切图是分开的）。项目一般是以 HTML 源文件的形式移交的，大部分设计师还会提供一些 PSD 源文件，方便日后修改。

量化投资专家系统移交的 HTML 页面效果图可能如图 2-46~图 2-50 所示。

金融专家系统 Financial expert system

专家 账户名

记住用户名和密码 忘记密码

登录

图 2-46 登录页面

角色权限设置

已有角色 19 个角色

客服经理 董事长 普通测试 会议管理 客服主管 理财顾问 总裁 CEO 风控总监 客服 专家 财务 超级测试员 超级管理员

客服经理 角色说明

后台菜单模块

- ☐ 添加菜单 ☐ 我的菜单 ☐ 添加会员 ☐ 员工管理
- ☐ 添加角色 ☐ 权限设置 ☐ 加管理员 ☐ 管理员表
- ☐ 流程管理 ☐ 添加流程 ☐ 添加插件 ☐ 插件管理
- ☐ 函数管理 ☐ 添加函数 ☐ 添加职位 ☐ 我的任务
- ☐ 系统维护 ☐ 修改密码 ☐ 磁盘管理 ☐ 高级搜索
- ☐ 知识库表 ☐ 导入知识 ☐ 任务轮训 ☐ 异地容灾
- ☐ 科目总账

后台隐藏模块

- ☐ 初始密码 ☐ 后台桌面 ☐ 编辑菜单 ☐ 更新菜单
- ☐ 编辑模块 ☐ 更新模块 ☐ 追加模块 ☐ 编辑专家

图 2-47 角色设置页面

写信 任务 (7) 消息 (7) 设置 已发送

标题

收件人 必须是用户名 多个用户可使用,分割

标题

正文

☐ 使用签名 ☐ 不使用签名

加入菜单 发送

图 2-48 发送页面

员工照片:

选择文件 未选择任何文件



姓名

性别

男性

生日

学历

大专

电话

住址

血型

入职

简介

专家级别

大专

☐ 禁用 ☐ 启用

存为模版

预览

保存

图 2-49 添加用户页面

所有员工

添加员工

ID	账户名称	真实姓名	开机时间	关机时间	操作
001	老徐	许阿婧	2001-12-05	2016-06-15	重置密码 编辑

1

1

2

3

4

5

6

7

下一页

共100页

图 2-50 列表页面

以列表页为例，设计师提交的文件一般包括 HTML 源文件、CSS 源文件、JavaScript 源文件、图片文件和 PSD 源文件。

列表页面主要由 HTML 头文件、外部 CSS、外部 JavaScript、公共头部、快捷菜单、左侧菜单、右侧子菜单、表格列表、分页符和版权信息这几个部分构成。列表页面的具体代码可能如下。

```

<!doctype xmlns="http://www.w3.org/1999/xhtml">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=
utf -8" />
<title>金融专家-列表页</title>
<!--引用外部CSS-->
<link rel="stylesheet" type="text/css" href="css/style.css">
<script type="text/javascript" src="JS/jquery.min.js"><!--调用
jquery--></script>
<script type="text/javascript" language="javascript">
//滚动公告js
</script>
</head>

<body>
<!--公共头部-->
<div id="header">
    <div class="top">
        <!--LOGO-->
        <h1>  </h1>
        <!--综合搜索框-->
        <form action="" method="get">
            <div class="top_serch">
                <select name="" class="input_select">
                    <option>专家</option>
                    <option>员工</option>
                    <option>角色</option>
                </select>
                <input name="" type="text" class="input_text" />
                <input name="" type="button" class="input_btn" />
            </div>
        </form>
        <!--滚动消息列表-->
        <ul class="scroll" style="margin-left:0; height:24px;
overflow:hidden;">

```

```

        <li><a href="#">量化投资专家系统发布</a></li>
        <li><a href="#">张三请你重置密码</a></li>
    </ul>
</div>
<!--快捷菜单-->
<div class="menu">
    <ul>
        <li class="on"><a href="index.html">管理员</a></li>
        <li> <a href="专家管理.html">专家系统 </a> </li>
        <li> <a href="#">统计分析 </a> </li>
        <li><a href="消息任务.html">消息任务 </a> </li>
        <li><a href="#">账户管理 </a> </li>
        <li><a href="#">常用窗口</a></li>
    </ul>
</div>
</div>
<div id="cont">
    <!--左侧菜单-->
    <div class="cont_left">
        <ul>
            <li class="on"> <a href="index.html">员工管理</a> </li>
            <li><a href="专家管理.html">专家管理</a></li>
            <li><a href="角色管理.html">角色管理</a></li>
            <li><a href="#">模块管理</a></li>
            <li><a href="#">日志管理</a></li>
            <li><a href="#">数据维护</a></li>
            <li><a href="#">系统维护</a></li>
            <li><a href="#">统计分析</a></li>
            <li><a href="消息任务.html">消息</a></li>
            <li><a href="#">账户</a></li>
        </ul>
        <div class="clear"></div>
    </div>
    <!--右侧工作区-->
    <div class="cont_right">
        <!--右侧工作区、子菜单部分-->

```

```

<ul class="title">
  <li>
    <select name="" class="add_select">
      <option>所有员工</option>
      <option>离职员工</option>
      <option>在职员工</option>
    </select>
  </li>
  <li class="l1"> <a href="添加员工.html">添加员工</a>
</li>
</ul>
<!--右侧工作区、列表部分-->
<table width="1020" cellspacing="0" class="staff_title" >
  <tr>
    <td width="170">ID</td>
    <td width="170">账户名称</td>
    <td width="170">真实姓名</td>
    <td width="170">开机时间</td>
    <td width="170">关机时间</td>
    <td width="170">操作</td>
  </tr>
</table>
<table width="1020" cellspacing="0" >
  <tr>
    <td width="170">001</td>
    <td width="170">老徐</td>
    <td width="170">徐志</td>
    <td width="170">2001-12-05</td>
    <td width="170">2016-08-15</td>
    <td width="170"><a href="#">重置密码 </a> <a
href="#">编辑</a></td>
  </tr>
</table>
<!--右侧工作区、分页部分-->
<div class="bottom_fy">
  <ul>

```

```

        <li><a href="#">1</a></li>
        <li><a href="#">1</a></li>
        <li><a href="#">2</a></li>
        <li><a href="#">3</a></li>
        <li><a href="#">4</a></li>
        <li><a href="#">5</a></li>
        <li><a href="#">6</a></li>
        <li><a href="#">7</a></li>
        <li class="l1">下一页</li>
        共<span class="span_1"><a href="#">100</a>
</span>页

    </ul>
</div>
<div class="clear"></div>
</div>
<div class="clear"></div>
<!--右侧工作区、版权部分-->
<div class="old">北京 软件有限公司 Copyright@2018</div>
</body>
</html>

```

至于外部调用的 Style.css 文件，设计师的主要设计思路可能如下。

(1) 尽量使用原生 HTML 标签来设计对应的页面元素，方便语义理解，这样设计师在元素初始化时，就会取消常用元素的基本样式，如 body、div、dl、dt、dd、ul、ol、li、h1、h2、h3、h4、h5、h6、pre、form、fieldset、input、textarea、blockquote、p{ margin:0; padding:0;}。

(2) 设置一些公共的样式，比如默认文本、对齐方式和超链接等。

接下来设计师一般会根据页面布局设置每个部分的具体样式，比如头部样式、菜单样式、标题样式和分页样式等。

如果某些页面有特殊设计最好单独拿出来，比如侧重于基本表单风格的添加员工页面样式，包括多选按钮的样式、角色权限设置页面的样式和发送页面的样式。

除此之外，设计师一般还会把登录页面的样式单独拿出来，这也符合登录页面的独立性。

登录页面中可能涉及图片目录，如图 2-51 所示。

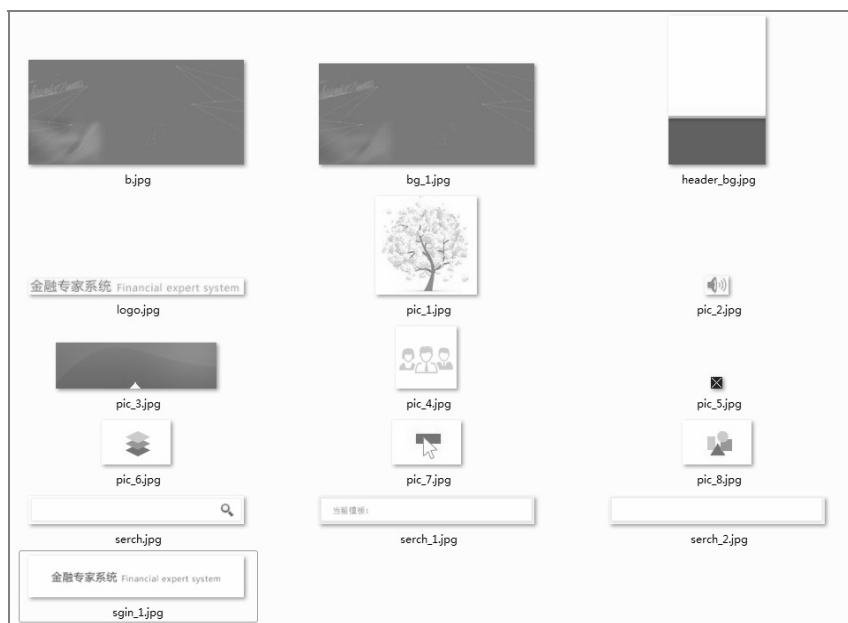


图 2-51 登录页面中可能涉及的图片

另外需要特殊说明的是，如果设计师移交的 CSS 文件是 GBK 编码形式的，则可以使用记事本打开，另存为 UTF-8 格式即可。

JavaScript 部分一般属于前段工程师的工作，设计师只要实现了基本的功能就可以了。

3

第 3 章

软件开发

编程时使用什么编程语言并不是特别重要，编程语言更多的是一种实现产品功能的工具，熟悉哪种编程语言就可以用哪种编程语言来开发，Java、C#、Python 和 PHP 都可以。本书侧重编程思想的介绍与说明，由于本书采用 B/S 架构，所以选择 Web 开发中比较流行的 PHP 作为主要编程语言进行讲解。

至于具体项目，我们采用迭代的方式进行开发，先开发简单的、周期短的，再开发复杂的、周期长的。这样发现错误可以及时改正，减少损失，毕竟设计费用远远低于工程造价费用。还有一种方式就是先核心再完善，如果公司技术实力非常强，就可以采用这种方式。例如，先开发一个模拟交易软件，然后开发策略，最后再完善其他功能。

本书考虑大部分读者的情况，决定先从简单的小功能开始开发。

开发时界面布局和功能参考原型图，具体样式参考网页设计，实现流程参考模块开发说明，需要调动的数据参考数据库设计，部门人员协同图如图 3-1 所示。

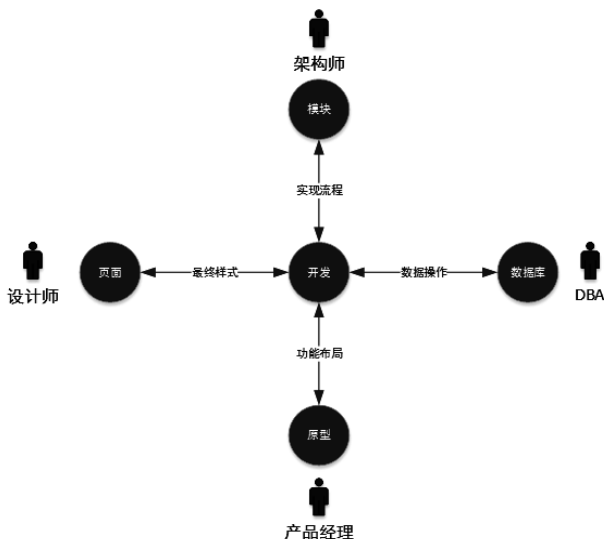


图 3-1 部门人员协同图

3.1 基本框架

基本框架可以使用已有的 PHP 框架，比如 Laravel 或者 Yii，也可以创建一个精简框架。由于本书更多的以原理实现功能为主，所以为了精简代码，特地采用极其简单的框架作为系统编程的主框架。

自己写一个简单框架，还是非常容易的。我们知道，一般的框架都是 MVC 模式的，M 是模型(Model)，V 是视图(View)，C 是控制器(Controller)。这里只做 MV 模式就足够用了，模型既是控制器，也是业务逻辑和数据操作，视图负责展示，最终以 HTML 的方式呈现给用户，如图 3-2 所示。

统一入口是为了方便权限管理、进行 URL 传值，入口文件默认是 index.php。

URL 传值主要包括具体的模型名、文件名和方法名三个部分。模型名和文件名构成了一个访问路径，方法名指定了调用的方法，比如“index.php?m=admin&c=index&a=main”是指调用 admin 文件夹下的 index.php 文件中的 main 方法。其中，m 表示文件夹名，c 表示文件名，a 表示方法名。

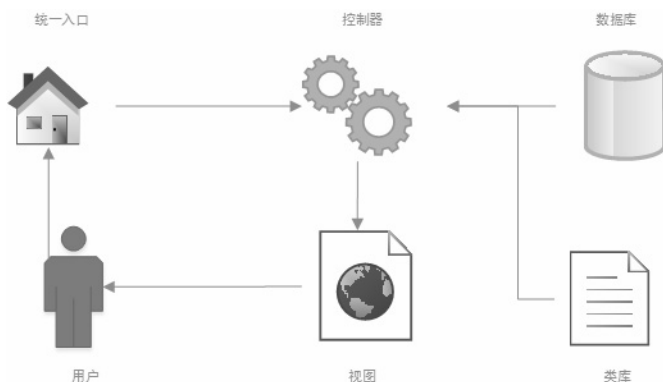


图 3-2 框架示意图

URL 规则制定好之后，我们就可以通过\$_GET 的方式进行调用了，获取 GET 传值的方法如下：

文件夹名 = \$_GET['m'];

文件名 = \$_GET['c'];

方法名 = \$_GET['a'];

一般情况下，为了文件管理方便，可以把所有的模型都放到一个叫作 m 的文件夹下，如果目录和文件不存在，则进行错误提示或访问默认文件。至于方法，大部分的框架都是面向对象的，可以简单到通过 include 直接调用，比如如下代码。

```

<?php
// 获取 GET 传值
if(@$_GET['m'] && @$_GET['c']){
    if(@$_GET['a']){
        // 判断是否是 PHP 已有函数
        if(function_exists($_GET['a'])){
            //提示函数已经存在
            echo 'function exists';
        }else{
            //调用文件
            include 'm/' . $_GET['m'] . '/' . $_GET['c'] . '.php';
            //调用方法
            $_GET['a'] ();
        }
    }
}
    
```

```

    }
}else{
    //调用文件
    include 'm/' . $_GET['m'] . '/' . $_GET['c'] . '.php';
    //调用默认方法
    main();
}
}else{
    //调用默认文件
    include 'm/index.php';
    //调用默认方法
    main();
}
?>

```

怎么样，是不是很简单？下面在 `m` 文件夹下添加一个叫作 `index.php` 的文件，然后编写一个叫作 `main()` 的函数，就可以实现最简单的 MV 模式了，代码如下。

```

m/index.php
<?php
function main()
{
    echo 'hello MV';
}
?>

```

这时再输入 `index.php` 就会显示“hello MV”字样。对于 MV 模式中的视图部分，为了管理方便一般会单独创建一个文件夹 `v`，为了方便系统维护，建议视图文件夹的名称和模型文件夹的名称一致。比如，在 `m/admin/index.php` 中创建一个叫作 `test` 的方法，然后对应创建一个 `v/admin/index_test.html` 的视图，代码如下。

```

m/admin/index.php
<?php
function test ()
{
    // include 方式，加载模板
    include 'v/admin/ index_test.html';
}

```

```
}
?>
v/admin/index_test.html
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>MV 测试</title>
</head>

<body>
欢迎光临!
</body>
</html>
```

上面的例子很简单，不过还需要完善两处能够更好地进行下一步编程。第一个需要完善的地方是，为了保证统一入口文件，需要对用户可能访问的文件做一个跳转或规避，比如如下代码。

```
Index.html
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<!--跳转-->
<meta http-equiv="Content-Type" content="text/html; charset=
utf-8" />
<title></title>
<meta http-equiv="refresh" content="0;url=index.php" />
</head>

<body>
</body>
</html>
```

另一个需要完善的地方就是配置文件，即便是这样最简单的 MV 框架，配置文件也是很重要的，因为有很多设置在后续开发中都能用到，比如连接数据库，以及数据库的“增、删、改、查”和代码的相关调试等。

这里把配置文件 `config.php` 放到一个叫作 `base` 的目录下，同时这个目录还包括一个叫作 `oa.php` 的数组文件。`config.php` 文件代码如下。

```

base/config.php
<?php
// UTF-8 编码
header("Content-type: text/html; charset=utf-8");
// 数据库配置和连接, 这里采用数组的模式
$data = array (
    'hostname' => 'localhost',
    'database' => 'ai_admin',
    'username' => 'root',
    // 'password' => '!@#abc123',
    'password' => '',
    'charset'  => 'utf8',
    'type'     => 'mysql',
    'debug'    => true,
    'pconnect' => 0,
    'autoconnect' => 0
);
//使用 MySQLi 的方式连接数据库, 当然也可以使用 PDO 或 MySQL
$MySQL = new
mysqli($data['hostname'],$data['username'],$data['password']);
$MySQL->select_db($data['database']);
$MySQL->set_charset($data['charset']);
//数据库连接错误处理
if($MySQL->connect_errno){
    //将数据库连接错误的信息写到 log.txt 日志中并退出系统
    file_put_contents('log.txt', date('Y-m-d H:i:s'). ' 数据库连
接失败'."\r\n", FILE_APPEND);
    //数据库连接错误
    exit();
}

// 开启 session
session_start();
//关闭错误调试
error_reporting(0);
//设置北京时间
date_default_timezone_set('PRC');
```

```
// 数据库常用函数
// 验证指定数据是否存在
function checking($s,$mysql){
    $q = $mysql->query($s);
    if($q){
        //返回数量
        return mysqli_num_rows($q);
    }else{
        return 0;
    }
}

// 返回查询后的数组
function DataList($s,$mysql)
{
    $q = $mysql->query($s);
    $data = array();
    if($q){
        //遍历数组
        while($row = mysqli_fetch_assoc($q)) {
            $data[]=$row;
        }
    }else{
        $data = 0;
    }
    //返回数组
    return $data;
}
?>
```

config.php 文件还可以包括添加日志函数和 post()、get()传值预处理函数等。config.php 文件设置好后，需要在同一个入口文件 index.php 顶部进行引用，代码如下。

```
require 'base/config.php';
```

一般在配置文件后，还会添加一个 APP_PATH 常数代替根目录，代码如下。

```
define('APP_PATH', __DIR__ . '/');
```


在项目的开发过程中，还会在一些表单页模块中遇到与表单操作相关的函数，这时可以把这些常用的表单函数当作一个第三方文件库存储到 `include` 目录中，比如用于数组生成表单的 `form.php`，代码如下。

```
Include/ form.php
<?php
// 生成 select 表单
function select($name,$d,$a)
{
    $s = "<select name='$name'>";
    if(is_array($a)){
        foreach($a as $k=>$v){
            if($d == $k){
                $s .= "<option value='$k' selected='selected'>$v
</option>";
            }else{
                $s .= "<option value='$k'>$v</option>";
            }
        }
    }
    $s .= "</select>";
    echo $s;
}
// 生成选项表单
function htmloption($d,$a)
{
    $s = "";
    if(is_array($a)){
        foreach($a as $k=>$v){
            if($d == $k){
                $s .= "<option value='$k' selected='selected'>$v
</option>";
            }else{
                $s .= "<option value='$k'>$v</option>";
            }
        }
    }
    echo $s;
```

```

    }
    // 指定数组生成 option 表单
    function arrayoption($key,$value,$d,$a)
    {
        $s = "";
        if(is_array($a)){
            foreach($a as $v){
                if($d == $v[$key]){
                    $s .= "<option value='$v[$key]' selected='selected'
>$v[$value] </option>";
                }else{
                    $s .= "<option value='$v[$key]'>$v[$value]</option>";
                }
            }
        }
        echo $s;
    }

    // 指定数组生成 select 表单
    function arrayselect($name,$key,$value,$d,$a)
    {
        $s = "<select name='$name'>";
        if(is_array($a)){
            foreach($a as $v){
                if($d == $v[$key]){
                    $s .= "<option value='$v[$key]' selected='selected'
> $v[$value] </option>";
                }else{
                    $s .= "<option value='$v[$key]'>$v[$value]</option>";
                }
            }
        }
        $s .= "</select>";
        echo $s;
    }

    // 生成 radio 表单
    function radio($name,$d,$a)

```

```

{
    $s = "";
    if(is_array($a)){
        foreach($a as $k=>$v){
            if($d == $k){
                $s .= "<label><input type='radio' value='$k'
name='$name' checked='checked' />$v</label>";
            }else{
                $s .= "<label><input type='radio' value='$k'
name='$name' />$v</label>";
            }
        }
    }
    echo $s;
}
?>

```

简单的 MV 框架就介绍到这里，后面随着项目的开发还可能对框架做一些完善。

3.2 登录模块

我们回过头来看一下模块开发说明中的登录模块流程图。登录模块是一个非常独立的模块，而且该系统需要登录后才能使用，所以登录模块更加独立。登录模块需要有几个功能点，其解决办法如下。

(1) 本地保存密码是使用 jquery.cookie.js 完成的，如果 cookie 存在就自动填充。

(2) 客户端验证码图片是调用 images/gifcode.php 文件来实现的，gifcode.php 文件主要调用 imagegif 函数来实现，用 SESSION 保存随机字符串。另外，为了让验证码更加美观，在 gifcode.php 目录中添加了一个叫作 youyuan.ttf 的字体库。

(3) 验证码的定时刷新使用 JavaScript 的 setInterval 方法，定时刷新验证码图片的 src 地址。

(4) 服务端验证用户是否登录成功需要调用数据库, 根据用户名返回这个用户的随机因子和“(密码+随机因子) MD5”加密后的字符串。如果客户端提交的密码和该用户的随机因子 MD5 加密后相等, 就表示登录成功, 分别生成\$_SESSION['uid']、\$_SESSION['utype']和\$_SESSION['uname']等。\$_SESSION['uid']用于保存用户的 ID, 系统数据大多是用用户 ID 进行关联的。\$_SESSION['utype']是为了表明系统是不是管理员, 因为系统采用统一登录界面, 所以为了防止歧义, 用\$_SESSION['utype']的值进行区别, 比如管理员是 0, 员工是 1。服务端验证的核心代码如下。

```
m/user/login.php
<?php
include 'include/box.php';
.....
```

上述代码引用了用于提示和跳转的 JavaScript 窗口文件 box.php, 它很简单, 其实就是输出一个 JavaScript 文本, 代码如下。

```
<?php
function alert($str,$url) {
    echo "<script type='text/javascript'>alert('" . $str .
    "');window.location.href ='" . $url . "';</script>";
}

// 登录验证
function login()
{    //账号和密码错误次数, 用于提示客户端是否显示验证码及锁定账号
    if($_SESSION['error']){
        //如果账号和密码错过一次, 就需要判断验证码是否相等
        if($_SESSION['randcode'] != $_POST['verification'])
        {
            // 提示验证码错误
            alert('验证码错误!', 'login.html');
        }else{
            // 查询账号和密码
            $type      = $_POST['user_type'];
            $username  = $_POST['user_name'];
            $password  = $_POST['user_password'];
```

```

        $memory = $_POST['memory_password'];
        user_login($type,$username,$password,$memory);
    }
}else{
    // 查询账号和密码
    $type = $_POST['user_type'];
    $username = $_POST['user_name'];
    $password = $_POST['user_password'];
    $memory = $_POST['memory_password'];
    user_login($type,$username,$password,$memory);
}
}

```

登录验证可以写成一个函数，这个函数由用户类型、用户名、密码和 cookie 状态 4 个参数构成。函数类型默认为 admin；cookie 状态默认为 ok，意思是需要保存账号和密码。

```

function user_login($type="admin",$username="", $password="",
$memory="ok"){
    //根据用户类型选择数据库，默认账号和密码不变
    global $MYSQL;
    switch($type){
        case "admin":
            // 后台管理员
            $MYSQL->select_db("ai_admin");
            $pa = DataList("SELECT `userid`,`password`,
`encrypt`,`roleid` FROM `admin` WHERE `username`= '$username' AND
`disabled`=0 LIMIT 1",$MYSQL);
            // 用户是否存在
            if($pa[0]['password'] != md5($password . $pa[0]
['encrypt']))){
                $_SESSION['error'] = 1;
                alert('管理员账号或者密码错误 ! ','login.html');
            }else{
                // 登录成功
                $_SESSION['uid'] = $pa[0]['userid'];
                $_SESSION['roleid']= $pa[0]['roleid'];
                $_SESSION['utype'] = 0;
                $_SESSION['uname'] = $username;
            }
        }
    }
}

```

```

        //客户端是否保存密码
        if($memory == "ok"){
            // 客户端默认两周
            $time = time()+2*7*24*3600;
            setcookie("username",$username,$time);
            setcookie("password",$password,$time);
        }else{
            //移除客户端 cookie
            $time = time()-100;
            setcookie("username","", $time);
            setcookie("password","", $time);
        }
        // 更新登录状态并跳转到后台主页
        header("Location: index.php?m=admin&c=index&a=
main");
    }
    break;
    case "member" or "expert":
        $MYSQL->select_db("ai_member");
        ..... 和管理管登录相似，只不过查询的是 ai_member 数据库.....
    break;
}
}
}

```

退出登录代码比登录代码要简单得多，只要清除对应的\$_SESSION['uid']，然后跳转到登录页面就可以了，退出登录代码如下。

```

function logout() {
    //清除 SESSION
    unset($_SESSION['uid']);
    unset($_SESSION['uname']);
    echo '<script type="text/javascript">
        window.top.location.href="login.html";
    </script>';
    die();
}

```

考虑到系统的基本权限，需要判断这个用户是否登录，只有登录之后才可以使用。最简单的方法就是判断这个用户的\$_SESSION['uid']是否存

在，如果不存在就要重新登录。为了避免每次都做判断，可以加载一个名叫 islogin.php 的文件来完成这个工作，代码如下。

```
islogin.php
<?php
error_reporting(0);
if(!$_SESSION['uid']){
    echo '<script type="text/javascript">
        window.top.location.href="login.html";
    </script>';
    die();
}
?>
```

3.3 会员管理

会员管理是管理系统的一个基本功能，一般分为固定会员管理和动态会员管理。固定会员管理如超级管理员的管理。超级管理员往往是必不可少的，一般这类会员我们直接在数据库上添加一条记录，如图 3-3 所示。



userid	username	password	roleid	encrypt	lastloginip
1	dong	cff736fe8d1505740b9d1779d056f078	1	5305	127.0.0.1

图 3-3 超级管理员记录

这里需要说明的是，为了保护用户的密码，其 password 字段是用户真实密码和加密字段的 MD5 值。

计算代码如下：

```
echo 'cff736fe8d1505740b9d1779d056f078' == md5('123456'. '5305');
```

一般来说，只要这个加密字段的字符串够大，是很难暴力破解的。当然，这里只有 4 个随机数字，还是相对容易破解的。

除超级管理员外，其他的管理员尤其是会员都是动态会员，很多会员往往就是一名真实的员工，员工不确定，会员也就不确定，因此需要有一

个能够查询、添加、修改会员的页面。

查询的代码比较简单，比如查询最后 50 名管理员的代码如下。

```
$MYSQL->select_db("ai_admin");
function adminlist(){
    global $MYSQL;
    //模块列表排序
    $user_data = DataList("SELECT * FROM `admin` ORDER BY
`userid` DESC LIMIT 0,50",$MYSQL);
    //管理员列表页
    include 'v/admin/admin_list.html';
}
```

添加会员可能稍微复杂一点，因为中间既牵扯到密码初始化的工作，又可能因为展现出现表单修改的问题，所以我们在添加会员这部分主要做以下三个工作。

首先，做一个通用的添加函数，这个添加函数可以自动识别表名、字段名和字段值。为了方便后台获取，在前台尽量让表单名和数据库的字段名一致，同时通过 GET 传值的方式获得需要操作的表名，代码如下。

```
function insert(){
    global $MYSQL;
    // 获得表名
    $t = $_GET['t'];
    //提示字符串
    $txt = '添加成功';
    // 根据表名设置需要转换的数据
    switch($t){
        case 'admin':          // 管理员表
            .....
            break;
        case 'admin_menu':    // 管理员菜单表
            break;
        .....
        default:
            die();
    }
    // 开始编写 SQL 语句
```



```

$sql = "INSERT INTO `{$t}`(";
$key = '';
$val = '';
    //遍历、传值
foreach($_POST as $k=>$v)
{
    $key .= '`' . $k . '`,';
    //为了安全需要进行转义
    $val .= "'" . addslashes(trim($v)) . "',";
}
    //去掉最后一个“, ”号
$key = substr($key, 0, -1);
$val = substr($val, 0, -1);
    //组合成 INSERT 语句
$sql = $sql . $key . ") VALUES(" . $val . ")";
    //执行 INSERT
if($MYSQL->query($sql)){
    //成功
}else{
    //失败
    echo $sql;
}
}

```

其次，写一个随机字符串生成函数，作为加密的随机因子（encrypt 字段），而且这个随机字符串的长度还可以设置，代码如下。

```

function getRandChar($length){
    //初始化字符串
    $str = '';
    //备选字符串，也可以有汉字等字符
    $strPol = "0123456789abcdefghijklmnopqrstuvwxyz";
    $max = strlen($strPol)-1;
    //循环抽取
    for($i=0;$i<$length;$i++){
        $str.=$strPol[rand(0,$max)];
    }
    //返回字符串
    return $str;
}

```

最后添加会员，仍然使用前面提到的 INSERT 函数，表单采用 POST 方式传值。核心代码如下。

```
function insert(){
    global $MYSQL;
    // 获得表名
    $t    = $_GET['t'];
    //提示字符串
    $txt = '添加成功';
    // 根据表名设置需要转换的数据
    switch($t){
        case 'admin':    // 管理员表
            //返回密码
            $ps          = $_POST['password'];
            //生成随机因子
            $encrypt     = getRandChar(4);
            //生成加密后的 32 位密码串
            $_POST['password']= md5($ps . $encrypt);
            //增加一个客户端没有提交的 encrypt 变量，便于后面 POST 方式
            的表单遍历
            $_POST['encrypt'] = $encrypt;
            break;
            .....
        }
        .....
    }
```

添加用户时为了保证用户名的唯一，客户端需要用 Ajax 方式验证用户名是否存在，为了方便维护，我们专门为 Ajax 调用编写了一个模块，这个模块就是 m/ajax/index.php。比如，管理员用户名是否被占用的代码如下。

```
m/ajax/index.php
<?php
.....
function checkadmin()
{
    global $MYSQL;
    $MYSQL->select_db("ai_admin");
```

```
// 默认名称的 GET 请求
$username = $_GET['username'];
$count = checking("SELECT `userid` FROM `admin` WHERE
`username` = '$username' LIMIT 1", $MYSQL);
echo $count;
}
.....
?>
```

添加用户后，用户的编辑就比较简单了，编辑时一般根据用户 ID 进行查询和修改。因为编辑操作是一个常用功能，所以也可以把编辑操作写成类似 INSERT 这样的通用函数 EDIT，编写时注意 WHERE 条件即可。

刚才添加了一名管理员，添加前台会员如员工、专家或交易员也是一样的方法。添加会员功能的前台页面没有几项内容，只是专家比员工多了一个级别而已，这个级别字段在 expert_data 表的 level 字段中，expert_data 实际上是 member 的专家级别扩展表。员工页面与专家页面的对比如图 3-4 所示。

会员类型 员工 ▼

账户名称 账户名

设置密码

确认密码

用户昵称

☐ 禁用 ☒ 启用

会员类型 专家 ▼

账户名称 账户名

设置密码

确认密码

用户昵称

专家级别 无级 ▼

☐ 禁用 ☒ 启用

图 3-4 员工页面与专家页面的对比

3.4 动态权限管理

与会员是动态的一样，很多员工和新增加的工作都拥有不同的使用权限，这些员工和新工作虽然权限各不相同，但是从统计上来看功能都是非

常相似甚至一致的，最终的体现是不同功能模块的组合。因此，我们设计了一套动态权限管理系统，这个系统主要包括模块管理（插件管理）、角色管理、权限设置和 URL 验证。

1. 模块管理

模块管理实际上是模块的“增、改、查”，而模块作为权限的最小单元，实际上还可以包含很多集成的方法，只不过这些集成的方法都属于一个权限组。从数据表 `admin_module` 可以知道，模块非常核心的功能是模块名称和 URL，模块名称用于默认显示，URL 则是权限的具体依据。

模块的“增、改、查”实现起来比较简单，模块列表的最终显示页可能如图 3-5 所示。

模块ID	名称	简介	路径	时间▼	作者▼	序号▼	操作
1	添加会员	添加会员，账户密码	<code>index.php?n=user&c=index&a=adduser</code>	2013-06-01	王昭东	81	编辑
2	员工管理	管理普通会员，不含交易员和专家	<code>index.php?n=user&c=index&a=userlist</code>	2013-06-01	王昭东	80	编辑
3	专家管理	专家管理	<code>index.php?n=user&c=index&a=expertlist</code>	2013-06-01	王昭东	79	编辑
4	角色管理	权限角色列表	<code>index.php?n=admin&c=index&a=rolalist</code>	2013-06-01	王昭东	78	编辑
5	插件管理	已经导入的文件	<code>index.php?n=file&c=index&a=filelist</code>	2013-06-01	王昭东	68	编辑
6	模块管理	显示模块列表	<code>index.php?n=admin&c=module&a=modulelist</code>	2013-06-01	王昭东	73	编辑
7	添加模块	添加模块	<code>index.php?n=admin&c=module&a=addmodule</code>	2013-06-01	王昭东	72	编辑
8	系统日志	日志，系统日志列表	<code>index.php?n=log&c=index&a=loglist</code>	2013-06-01	王昭东	0	编辑
17	系统维护	系统维护主界面	<code>index.php?n=system&c=index&a=main</code>	2013-06-08	王昭东	0	编辑
18	统计分析	统计分析主界面	<code>index.php?n=count&c=index&a=main</code>	2013-06-08	王昭东	0	编辑

图 3-5 模块列表的最终显示页

这里有一点需要注意的是，由于 URL（路径）字段中包含 `&` 符号，所以在客户端显示时经常使用 `htmlspecialchars` 函数进行转义。

2. 角色管理

角色管理也比较简单，首先添加一个角色名称，然后进行保存，保存后数据库 `admin_role` 表中就有了一个对应的角色 ID（`roleid`），接着就可以通过勾选的方式来编辑这个角色所对应的模块，如图 3-6 所示。

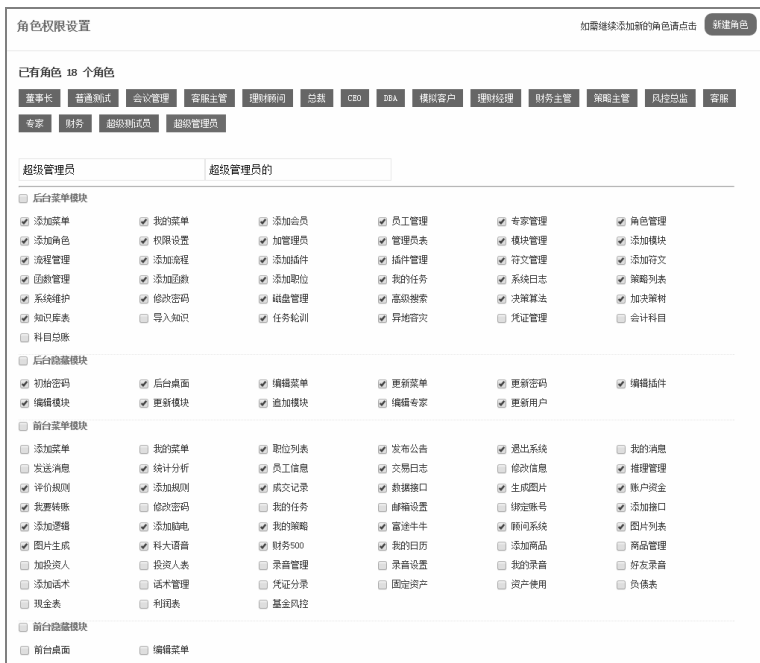


图 3-6 角色管理页面

为了方便管理员选择模块,我们根据表 `admin_module` 中的 `type` 用户类型字段 (0 为会员, 1 为管理员) 和 `show` 是否显示到菜单中的字段 (0 为显示, 1 为不显示) 分成了 4 组模块。例如, 后台菜单前 100 个模块的代码如下。

```
$module_admin_data = DataList("SELECT `moduleid`,`module`,`description` FROM `admin_module` WHERE `disabled`=0 AND `show`='1' AND `type`='1' ORDER BY `listorder` DESC LIMIT 0,100",$MYSQL);
```

有了角色和模块表, 接下来只要关联到相应的角色字段就可以了。例如, 关联 `admin` 表中的 `roleid` 和 `member` 表中的 `roleid`。

3. 权限设置

权限设置也因此变得非常简单, 找到对应的会员, 然后通过“权限设置”下拉列表框选择对应的角色, 设置好后通过 Ajax 更新相应表单的 `roleid` 字段就可以了。另外, 为了方便管理管理员的权限, 可以在页面左上角的下拉列表框中选择“管理员”, 如图 3-7 所示。

加入菜单	管理员	请输入用户名	搜索
会员	管理员		
用户名	真实姓名	权限设置	
dong	王昭东	超级管理员	
test	测试用户1	超级测试员	
wang	测试用户2	财务	
cenziye	岑子哥	无	

图 3-7 权限设置页面

4. URL验证

URL 验证是一个非常有趣的问题，这个问题主要是为了防止员工盗用其他人的权限。

URL 验证加载在 index.php 入口文件中，上线后为了性能考虑，可以将模块表、权限表复制一份放到内存表中。修改后的 index.php 文件的代码如下。

```

index.php
<?php
define('APP_PATH', __DIR__ . '/');
require 'base/config.php';
if(@$_GET['m'] && @$_GET['c']){
    if(@$_GET['a']){
        if(function_exists($_GET['a'])){
            echo 'function exists';
        }else{
            //这里需要加一个权限验证，验证是否有这个模块的权限
            $URL = 'index.php?m=' . $_GET['m'] . '&c=' .
$_GET['c'] . '&a=' . $_GET['a'];
            //登录除外
            if($_GET['m'] != 'user' && $_GET['c'] == 'login'){
                try{
                    //异常处理
                    if($_SESSION['uid']){
                        //查询角色表
                        $roleid = $_SESSION['roleid'];
                        $role = DataList("SELECT `modules`
FROM `admin_role` WHERE `roleid`= '$roleid' LIMIT 1", $MYSQL);

```

```

        $modules= $role[0]['modules'];
        //查询模块
        $module = DataList("SELECT `descrip
tion`,`moduleid`,`module`,`url` FROM `admin_module` WHERE
`moduleid` IN($modules) LIMIT 0,100",$MYSQL);
        //echo $URL;
        //遍历模块是否包含当前 URL
        foreach($module as $v){
            if($v['url'] == $URL){
                // m 为文件夹, c 为文件名, a 为方法名, 其他参数也以 GET 方式传值
                //echo $v['url'];
                include 'm/' . $_GET['m'] . '/' .
$_GET['c'] . '.php';

                $_GET['a']();
            }
        }
    };
    }catch(Exception $e){
        //输出异常
        echo $e->getMessage();
    }
    }else{
        include 'm/' . $_GET['m'] . '/' . $_GET['c'] .
'.php';

        $_GET['a']();
    }
    //根据角色权限查找 URL 是否与当前 URL 匹配
    //如果匹配, 则执行; 否则提示没有权限, 停止运行, 返回上一级目录
    }
    }else{
        include 'm/' . $_GET['m'] . '/' . $_GET['c'] . '.php';
        main();
    }
    }else{
        include 'm/index.php'; main();
    }
    ?>

```

5. 插件管理

插件管理本质上属于模块管理的一部分，主要包括上传源码文件、解压缩文件、连接数据库和导入 SQL 文件，示例代码如下。

```
m/file/index.php
function inputplug()
{
    global $MYSQL;
    //默认第一步
    $step = $_GET['step']?$_GET['step']:0;
    //错误提示
    $err = '';
    switch($step){
        case 0:
            //第一步，基本信息，利用 SESSION 保存
            $_SESSION['username'] = post('username');
            $_SESSION['filename'] = post('filename');
            $step = 1;
            break;
        case 1:
            //第二步，配置文件，利用 SESSION 保存
            $_SESSION['ip'] = post('ip');
            $_SESSION['user'] = post('user');
            $_SESSION['password'] = post('password');
            $_SESSION['dataname'] = post('dataname');
            //验证数据库文件
            $con = @mysql_connect($_SESSION['ip'],$_SESSION
['user'],
            $_SESSION['password']);
            $db = @mysql_select_db($_SESSION['dataname'],
$con);

            if(!$con || !$db){
                $err = '数据库连接错误，请重新填写!';
                //返回上一步
                $step = 1;
            }else{
                $step = 2;
            }
    }
}
```



```

        @mysql_close($con);
    break;
    case 2:
        //第三步，上传文件
        $_SESSION['filepath'] = post('upfile');
        $input = post('input');
        //插件存放的目录
        $pathname=$path = 'plugin/' . $_SESSION['uid'] . '/' .
post('pathname');
        $_SESSION['pathname'] = $pathname;
        //判断是否为 ZIP 压缩，如果是 ZIP 压缩，则要解压缩
        if(post('type') == 'zip'){
            $zip = new ZipArchive();//新建一个对象
            if ($zip->open($_SESSION['filepath']) ==
TRUE) {

                $zip->extractTo($pathname);
                $zip->close();
                //删除文件
                unlink($_SESSION['filepath']);
            }
        }
        $step = 3;

```

SQL 文件的导入是本节的一个难点，可以把 SQL 文件的导入分成 4 个步骤。

- (1) 根据配置文件连接对应的数据库。
- (2) 返回上传的文件中所有的 SQL 文件对象并形成数组。
- (3) 遍历 SQL 文件，逐行读取，先去掉注释，再开始操作，操作的依据是“;”分隔符。
- (4) 安全起见，SQL 文件操作后需要删除。

导入 SQL 文件的代码如下。

```

//是否导入数据库
if($input){
//连接数据库
$con = new mysqli($_SESSION['ip'],$_SESSION['user'],

```

```

$_SESSION['password']);
    $con->select_db($_SESSION['dataname']);
    $con->set_charset('utf8');
    if($con->connect_errno){
        $err = '数据库连接错误, 请重新填写!';
        $step = 1;
    }else{
        //遍历 SQL 文件
        foreach (glob("$pathname/*.sql") as
$filename) {
            //逐行读取
            $templine = '';
            $lines = file($filename);
            $annotation = 0;
            foreach($lines as $line){
                //去掉注释
                if($line == '/*'){ $annotation = 1;}
                if($line == '*/'){ $annotation = 0;}
                if($annotation == 1){continue;}
                if(substr($line, 0, 2) == '--' ||
$line == ''){continue;}

                $templine .= $line;
                if (substr(trim($line), -1, 1) ==
';'){
                    //开始操作

                    if(!$con->query($templine)){ $err = 'Error performing query ' .
$templine . ': ' . $con->connect_errno . '<br />';}
                    $templine = '';
                }
            }
            //删除 SQL 文件
            unlink($filename);
        }
    }
    $con->close();
}
break;

```

```

case 3:
    //保存相关信息并退出
    $filename = $_SESSION['filename'];
    $filepath = $_SESSION['pathname'];
    $userid = $_SESSION['uid'];
    $uploadtime = time();
    $username = $_SESSION['username'];
    $uploadip = $_SERVER['REMOTE_ADDR'];
    $dataset = 'ip|' . $_SESSION['ip'] . ',user|' .
$_SESSION['user'] . ',password|' . $_SESSION['password'] .
',dataname|' . $_SESSION['dataname'];
    $disabled = 0;
    $sql"INSERT INTO `admin_files`(`filename`,
`filepath`,`userid`,`uploadtime`,`username`,`uploadip`,`dataset`,
`disabled`) VALUES('$filename','$filepath','$userid','$upload
time','$username','$uploadip','$dataset','$disabled');"
    if($MYSQL->query($sql)){
        //添加成功
    }else{
        //失败
        echo $sql;
        die;
    }
    $step = 4;
    break;
}
include 'v/file/file_add.html';
}

```

对于 v/file/file_add.html 视图页面,可以通过\$step 的值来确定当前步骤。

3.5 自定义菜单与快捷菜单

登录系统之后,会根据用户的权限返回其所有的功能模块。一般来讲,只返回那些前台可以显示的模块。比如,可以先查询用户的角色,角色在用户登录时会生成一个 SESSION,并使用\$_SESSION['roleid']这个变量来保

存用户的角色 ID，从而简化这个过程，代码如下。

```
m/ user/index.php
<?php
//返回用户基本信息
$user_info = DataList("SELECT * FROM `member` WHERE `memberid`='
$uid' LIMIT 1",$MYSQL);
//返回角色权限
$roleid = $user_info[0]['roleid'];
$role_data = DataList("SELECT `modules` FROM `admin_role`
WHERE `roleid`= '$roleid' LIMIT 1",$MYSQL);
//返回对应的模块列表
$modules = $role_data[0]['modules'];
$module_data = DataList("SELECT
`description`,`moduleid`,`module`,`url` FROM `admin_module` WHERE
`moduleid` IN($modules) AND `show`= 0 ORDER BY `listorder` DESC LIMIT
0,100",$MYSQL);
.....
?>
```

客户端可以通过 foreach 进行遍历，代码如下。

```
v/user/left.html
.....
<li class="on"> <a href="javascript:void(0)" onClick="role
('#role_all');">我的权限</a></li>
<li id="role_all" class="li">
<?php foreach($module_data as $v){ ?>
<a href="<?php echo $v['url'];?>" target="main" title="<?php
echo $v['description']; ?>"><?php echo $v['module'];?></a>
<?php } ?>
</li>
.....
<script type="text/javascript">
//显示隐藏列表
function role(str){
    $(str).toggle(500);
}
</script>
```

如果用户的权限非常大，可以操作 100 多个功能模块，那么用起来就

会不太方便，这时就需要为用户做一个自定义菜单模块。自定义菜单可以让用户方便地对所有权限进行管理。比如，把与消息相关的模块放在一起、把与交易相关的模块放在一起。

想要实现自定义菜单功能，需要开发两个小模块，一个是添加自定义菜单模块，另一个是自定义菜单管理模块，而菜单管理模块包含编辑菜单这个子功能。

添加自定义菜单模块的项目比较少，这里只做一级目录，也就是说没有上级菜单，默认全部都是一级菜单，添加自定义菜单页面如图 3-8 所示。

图 3-8 添加自定义菜单页面

添加自定义菜单页面主要是循环输出该用户的权限模块，仍然可以使用 `foreach` 进行遍历。为了传值方便，输入表单统一使用“ids[]”这个名称。另外，为了方便客户端选取，在输入外围添加了 `label` 标签，代码如下。

```
v/user/menu_add.html
.....
<! --用户权限列表 -->
<?php foreach($module_data as $v){ ?>
<label title="<?php echo $v['description'];?>">
<input name="ids[]" type="checkbox" value="<?php echo
$v['moduleid'];?>" />
<?php echo $v['module'];?> </label>
```

```
<?php }?>
```

.....

关于子菜单的排序,前端可以使用一个隐藏表单记录输入表单的选择顺序,若顺序是输入表单的值,则将这个表单直接代替 `ids[]` 表单就可以了。

服务端的获取相对容易一些,使用 `implode` 方法将所有的模块 ID 组成字符串,存入数据库 `member_menu` 表的 `modules` 字段中,或者存入 `admin_menu` 表的 `modules` 字段中。添加或修改可以使用通用的 `INSERT` 或 `EDIT` 函数来实现。`implode` 方法的代码如下。

```
m/user/index.php
```

```
$_POST['modules']= implode(', ', $_POST['ids']);
```

菜单列表就比较简单了,主要是查询和显示功能。编辑菜单和添加菜单的页面与功能相似,只是程序处理上一个添加(可以使用 `INSERT` 函数),另一个是修改(可以使用 `EDIT` 函数)。

有了自定义菜单,当用户进入默认主界面时,就可以个性化地现实自定义菜单了。比如,默认员工最多可以查询 30 个菜单,代码如下。

```
m/ user/index.php
```

.....

```
$MYSQL->select_db("ai_member");
```

```
$menu_data = DataList("SELECT * FROM `member_menu` WHERE  
`userid`= '$uid' AND `disabled`= 0 ORDER BY `listorder` DESC LIMIT  
0,30", $MYSQL);
```

.....

客户端在渲染时,可以是一个嵌套循环,其代码如下。

```
v/user/left.html
```

.....

```
<?php
```

```
foreach($menu_data as $vv){
```

```
?>
```

```
<ul>
```

```
<li class="on"> <a href="javascript:void(0)" onClick="role  
('#role_<?php echo $vv['menuid']; ?>');"><?php echo $vv['menuna  
me']; ?></a></li>
```

```
<li id="role_<?php echo $vv['menuid']; ?>" class="li">
```

```
<?php
    //将模块列表字符串分割成数组
    $mymodules = explode(',', $vv['modules']);
    //判断是否在已有限权数组中
    foreach($module_data as $o){
        if(in_array($o['moduleid'], $mymodules)){
            ?>
                <a href="<?php echo $o['url'];?>" target="main"
title="<?php echo $o['description']; ?>"><?php echo
$o['module'];?></a>
                <?php }
            } ?>
        </li>
    </ul>
<?php }?>
.....
```

自定义菜单的最终效果如图 3-9 所示。

我的权限

商品管理

我的账户

消息公告

客户管理

录音管理

话术管理

其他菜单

我的菜单

添加菜单

我的日历

加入菜单

全选

选类

全部执行

菜单名称	菜单说明	菜单排序	模块集合	操作
商品管理	金融产品管理	10	82, 83	编辑
我的账户	账户密码	5	21, 37, 38, 46, 81	编辑
消息公告	消息公告管理	3	12, 13, 47, 48, 49, 102	编辑
客户管理	投保人管理	3	84, 85	编辑
录音管理	电话录音管理	2	86, 87, 88, 89	编辑
话术管理	话术管理	0	90, 91	编辑
其他菜单	其他菜单	0	37, 38, 81	编辑

图 3-9 自定义菜单的最终效果

在图 3-9 中，用户的权限包含了所有可以显示的模块，类似操作系统的所有程序。自定义菜单是对所有程序的二次整理，除左侧菜单外，还在顶部增加了一个快捷菜单的功能，这个快捷菜单类似操作系统底部的锁定任务栏。为了实现这个功能，我们在前端增加了一个公共的快捷菜单模板页面 `v/admin/shortcut.html`，用于判断当前 URL 是否在快捷菜单列表

member_menu_shortcut 表的 roles 字段中。roles 字段可以在用户初次进入默认页面时生成一个\$_SESSION['shortcut']并保存，代码如下。

```
m/ user/index.php
.....

//自定义快捷菜单

$data = DataList("SELECT * FROM `member_menu_shortcut` WHERE
`userid`= '$uid'  LIMIT 1",$MYSQL);

if(count($data)){
    //分割处理

    $shortcut = explode(',',$data[0]['roles']);
    $_SESSION['shortcut'] = $shortcut;
}
.....
```

如果有内容，就显示到前端快捷菜单中。快捷菜单一般放在顶部模板中，代码如下。

```
v/user/head.html
.....

<ul>
<?php
    foreach($shortcut as $v){
        $vv = explode('|',$v);
        ?>
        <li><a href="<?php echo $vv[1]; ?>" target="main"><?php
echo $vv[0]; ?> </a></li>
        <?php }?>
    </ul>
.....
```

公共的快捷菜单模板就是一个简单的按钮，嵌在所有可能作为快捷菜单的页面中，如果读者的按钮有两种样式，则可以再新建一个模板 2。模板 1 的代码如下。

```
v/admin/shortcut.html
<?php
//快捷菜单是否存在，如果存在就显示取消，否则就显示添加
$url = 'index.php?' . $_SERVER["QUERY_STRING"];
$isurl = 0;
```



```

foreach($_SESSION['shortcut'] as $v){
    $vv = explode('|',$v);
    if($vv[1] == $URL){
        $ISURL = 1;
        break;
    }
}
//按钮样式
if($ISURL){
    echo '<span onClick="delShortcut()">取消快捷</span>';
}else{
    echo '<span onClick="addShortcut()">加入菜单</span>';
}
?>

```

addShortcut()方法是用 Ajax 的方式添加的新的快捷菜单,其 JavaScript 代码如下。

```

v/user/g_head.html
.....
//追加快捷方式
addShortcut = function(){
    var name = prompt("请输入菜单名称:", "未定义");
    if(name != null){
        //返回当前 URL, 用 Ajax 方式提交
        var url = 'index.php' + window.location.search;
        $.get("index.php?m=user&c=index&a=addshortcut",{name:name
,url:url},function(result){
            if(result == 'ok')alert('自定义菜单添加成功,记得刷新主
页面哦!');
        });
    }
}

```

delShortcut()方法是用 Ajax 的方式取消已有的快捷菜单,其 JavaScript 代码如下。

```

delShortcut = function(){
    //返回当前 URL, 用 Ajax 方式提交
    var url = 'index.php' + window.location.search;

```

```
$.get("index.php?m=user&c=index&a=delshortcut",{url:url},
function(result){
    if(result == 'ok')alert('快捷菜单已经删除，记得刷新主页面哦!
');
    });
}
.....
```

服务端代码主要是 addShortcut 方法和 delShortcut 方法，默认不超过 10 个快捷菜单，如果超过就顺延删除第一个。同时，为了方便区分管理员和会员，快捷菜单可以通过 \$_SESSION['utype'] 变量进行区分，代码如下。

```
m/admin/index.php
.....
//添加管理会员快捷菜单
function addshortcut(){
    //返回客户端提交的数据
    $name= $_GET['name'];
    $url = $_GET['url'];
    $new = $name . '|' . $url;
    global $MYSQL;
    //查询这个用户的快捷菜单
    $uid = $_SESSION['uid'];
    $utype = $_SESSION['utype'];
    if($utype){//会员
        $MYSQL->select_db("ai_member");
        $data = DataList("SELECT `roles` FROM
`member_menu_shortcut` WHERE `userid`= '$uid' LIMIT 1",$MYSQL);
    }else{//管理员
        $MYSQL->select_db("ai_admin");
        $data = DataList("SELECT `roles` FROM
`admin_menu_shortcut` WHERE `userid`= '$uid' LIMIT 1",$MYSQL);
    }
    if(count($data)){
        //分割处理
        $shortcut = explode(',',$data[0]['roles']);
        //如果大于 9 个就删除最后一个
        if(count($shortcut)>9){
            array_pop($shortcut);
        }
    }
}
```

```

    }else{}
    //追加到第1个元素
    array_unshift($shortcut,$new);
    //数组元素转换成字符串
    $str = implode(',',$shortcut);
    //更新
    if($utype){//会员
        $sql = "UPDATE `member_menu_shortcut` SET `roles` =
'$str' WHERE `userid` = $uid LIMIT 1";
    }else{//管理员
        $sql = "UPDATE `admin_menu_shortcut` SET `roles` =
'$str' WHERE `userid` = $uid LIMIT 1";
    }
    if($MYSQL->query($sql)){
        OALog('','0','添加员工快捷方式','', '');
        echo 'ok';
    }
}
}else{
    //直接添加
    $str = $new;
    if($utype){//会员
        $sql = "INSERT INTO
`member_menu_shortcut`(`userid`,`roles`) VALUES($uid,'$str');";
    }else{
        $sql = "INSERT INTO
`admin_menu_shortcut`(`userid`,`roles`) VALUES($uid,'$str');";
    }
    if($MYSQL->query($sql)){
        OALog('','0','添加员工快捷方式','', '');
        echo 'ok';
    }
}
}
}

```

删除管理员、会员指定快捷菜单的操作与添加操作有点相似，首先需要返回客户端提交的数据，比如 URL、用户 ID 和类型，然后通过遍历 \$_SESSION['shortcut']数组的方法删除对应的 URL 字段，最后将删除后的

字符串通过\$_SESSION['utype'] (用户类型) 再次更新到数据库对应的表中, 代码如下。

```
//删除管理员、会员指定快捷菜单
function delshortcut() {
    global $MYSQL;
    //返回客户端提交的数据
    $url = $_GET['url'];
    $uid = $_SESSION['uid'];
    $utype = $_SESSION['utype'];
    //遍历并删除指定的 URL
    $arr = [];
    foreach($_SESSION['shortcut'] as $v) {
        $vv = explode('|', $v);
        if($vv[1] != $url) {
            $arr[] = $v;
        }
    }
    $str = implode(',', $arr);
    if($utype) { //会员
        $MYSQL->select_db("ai_member");
        $sql = "UPDATE `member_menu_shortcut` SET `roles` = '$str' WHERE `userid` = $uid LIMIT 1";
    } else { //管理员
        $MYSQL->select_db("ai_admin");
        $sql = "UPDATE `admin_menu_shortcut` SET `roles` = '$str' WHERE `userid` = $uid LIMIT 1";
    }
    if($MYSQL->query($sql)) {
        echo 'ok';
    } else {
    }
}
.....
?>
```

快捷菜单的最终效果如图 3-10 所示。

添加菜单		菜单列表	
我的权限	取消快捷	全选	选项
	全部执行		
	商品管理	菜单名称	菜单说明
	我的账户	商品管理	金融产品管理
	消息公告	我的账户	账户密码
客户管理	消息公告	消息公告	消息公告管理
	录音管理	客户管理	投资人管理
		菜单排序	10
			5
			3
			3

图 3-10 快捷菜单的最终效果

3.6 站内信与动态流程

聊天功能可以非常简单，也可以比较复杂。简单的聊天功能就像一个站内信，复杂的聊天功能就像一个动态的办公流程。

根据产品要求和模块设计，聊天功能共有三个版本，其中 V1.0 版本仅仅是一个消息发送和回复的站内信，达到最小可用即可，在技术实现上没有什么难度，只是在消息群发的时候需要循环增加收件人的用户名。与站内信一样简单的还有公告，公告一般是由后台管理员发布的，有一个起止时间，超过了结束时间就不再显示。通过 `member_announce` 表可知 `starttime` 和 `endtime` 都是数字格式的，针对客户端提交的日期格式的字符串，后端可以通过 `strtotime` 函数来转换。

关于邮件部分，我们还是比较倾向于第三方软件，比如 Outlook 或 Foxmail。

站内信（我的消息模块）中的邮件设置是为了使用已有的 E-mail 而做的一个 Web 应用，这个应用主要方便通过 SMTP 的方式发送邮件。我们知道 `message_set` 表中只有 `emailset` 一个字段，那么在做邮件设置时，需要把字符串按照键值对的方式组织起来，也可以使用函数直接将数组转换成字符串，如 `json_encode` 和 `json_decode`，当然也可以使用 `implode` 和 `explode` 这种更简单的方式，默认推荐 JSON 方式。另外需要说明的是，在使用

`json_decode` 函数且第二个参数为 `true` 时，可以直接转换成可用数组形式而非对象。

邮件设置的主要代码如下。

```
m/im/index.php
//获得邮箱设置信息
function getEmailSet(){
    global $MYSQL;
    $uid = $_SESSION['uid'];
    $data = DataList("SELECT * FROM `message_set` WHERE
`memberid`= '$uid' LIMIT 1",$MYSQL);
    $show = $data[0];
    $set = json_decode($show['emailset'],true);
    return $set;
}
```

发送邮件这里使用 **SMTP**，当用户选择发送邮件时，需要先判断是否进行过邮件设置，没设置过就直接跳转到设置页面，设置后再通过 **SMTP** 方式发送邮件。**PHPMailer** 有一个现成的 **SMTP** 发送类，这个类可以发送 **HTML** 文档和附件，**PHPMailer** 下载地址是 <https://github.com/PHPMailer/PHPMailer>。

其实，**PHPMailer** 文件一般只需要 `class.phpmailer.php` 和 `class.smtp.php` 两个类就可以了，如果还想做 **POP3D** 验证，也可以把 `class.pop3.php` 这个类复制到项目的 `include` 目录中。

为了让 **SMTP** 正常工作，需要做两个小的设置和改动，一个是服务端要开启 `openssl`，另一个是把 `class.phpmailer.php` 文件中的“`iso-8859-1`”换成“`UTF-8`”，代码如下。

```
//public $CharSet = 'iso-8859-1';
public $CharSet = 'UTF-8';
```

另外，现在很多邮箱服务器都需要在邮箱设置中开启 **SMTP**、**POP3** 或 **IMAP**，只有开通后才可以通过第三方客户端收发这个服务商的邮件，比如腾讯 **QQ** 邮件，就会通过手机短信验证的形式，发送给用户一个口令（授权码），这个口令就是邮件设置中的密码。对于多个收件人来说，可以通过

foreach 函数循环增加\$mail->addReplyTo 方法和\$mail->addAddress 方法,主要代码如下。

```
m/im/index.php
.....
//发送电子邮件
function emailsend(){
    $subject = $_POST['subject'];
    $content = $_POST['content'];
    $set = getEmailSet();
    $names = explode(';', $_POST['names']);
    send($set, $names, $subject, $content);
    back('已经发送! ', -1);
}
//发送邮件函数, 设置数组、收件人、标题、内容
function send($email, $to, $subject, $content){
    //导入 PHPMailer 的 SMTP 类
    require 'include/class.phpmailer.php';
    require 'include/class.smtp.php';

    //创建 PHPMailer 类
    $mail = new PHPMailer;
    //使用 SMTP 发送邮件
    $mail->isSMTP();
    //发送回执, 0 为关闭, 1 为客户端, 2 为服务端
    $mail->SMTPDebug = 0;
    //使用 HTML 格式
    $mail->Debugoutput = 'html';
    //服务端
    $mail->Host = $email['smtp'];
    //设置端口号 25、465、587、994、995 等, 默认为 25
    $mail->Port = 25;
    //认证
    $mail->SMTPAuth = true;
    //用户名
    $mail->Username = $email['email'];
    //密码或权限码
    $mail->Password = $email['password'];
```

```
// $mail->Password = "971050";
// 接收者
$mail->setFrom($email['email'], 'First Last');
// 群发邮件
foreach($to as $name){
    // 邮箱
    $mail->addReplyTo($name, 'fo');
    // 地址
    $mail->addAddress($name, 'to');
}
// 标题
$mail->Subject = $subject;
// HTML 源码
$mail->msgHTML($content);
$mail->AltBody = 'ss';
// 附件, 暂时不设置附件
// $mail->addAttachment('images/b.jpg');
// 发送
if (!$mail->send()) {
    echo "Mailer Error: " . $mail->ErrorInfo;
} else {
    echo "Message sent!";
}
}
```

相对发送邮件而言, 读取邮件虽然技术实现原理不难, 但是需要处理的页面相对多一些, 这里主要使用 IMAP 系列函数。另外一个需要注意的是, HTML 邮件需要独立的样式页面, 从而避免邮件的样式冲突, 代码如下。

```
m/im/index.php
// 最新邮件
function newemail(){
    // 永不超时
    // set_time_limit(0);
    $menu = 'newemail';
    $set = getEmailSet();
    // 这里使用 POP3 协议
    $imap = imap_open("{ " . $set['pop3'] . " : " . $set['pop3]
```



```
port'] . "/pop3/ssl/novalidate-cert}INBOX", $set['email'],
$set['password']);
    if($imap){
        $num = imap_num_msg($imap);
        $n = $_GET['n']?$_GET['n']:$num;
        if( $n >0 ) {
            //获取邮件正文
            $body = imap_qprint(imap_body($imap, $n));
            //相关处理
            .....
        }
        //关闭 IMAP
        imap_close($imap);
    }
    include 'v/im/email_new.html';
}
```

邮件属于过渡产品，未来将用智能聊天系统代替邮件的功能。如果从工作量的情况来考虑的话，则可以直接通过超链接的方式打开本地的邮件客户端，代码如下。

```
v/im/email_menu.html
.....
<a href="mailto:<?php echo $set['email']; ?>">客户端</a>
.....
```

3.7 任务模块

任务模块的任务来源主要由两部分构成，一部分是系统根据工作流程自动触发的，另一部分是管理员人为添加的。如果工作流程是静态的，则实现起来可能比较简单：针对某个用户的某个模块发起任务，一旦成功提交就发送给下一个人，同时产生一个任务消息。但是实际上工作流程可能随着时间的变化而变化，为了适应这种变化，不能每次都重新开发一个功能，所以开发了动态流程系统。

在动态流程中最好先完善一下组织机构，组织机构通过 `member_group`

表中的 pids 字段来实现多层级的关系，这里的 pids 字段使用的是 varchar 类型而不是 int 类型，主要是为了适应特殊组织结构而留出一些扩展，默认为树形组织结构，一般组织结构都会给出直接上级和所属员工（通过【添加】按钮可以有针对性地选择员工），所以通过组织结构可以很快地确定用户，通过用户的权限确定角色，再通过角色确定具体的模块，组织机构编辑页面如图 3-11 所示。

图 3-11 组织机构编辑页面

核心代码如下。

```
m/user/index.php
.....

function edit(){
    case 'member_group':// 更新职位
        //切换数据库
        $MYSQL->select_db("ai_im");
        $groupid= $_GET['groupid'];
        $where = "`groupid` = '$groupid'";
        $_POST['pids'] = implode(',',$_POST['pids']);
        //判断是否有主管用户名
        if(strpos($_POST['mainusername'],',')==0){
            $_POST['mainusername']= substr($_POST['mainu
            sername'],1);
```

```

};
//判断是否有员工用户名
if(strpos($_POST['members'],',')==0){
    $_POST['members']=substr($_POST['members'],1);
};
//跳转到组织机构列表页面
$url = 'index.php?m=im&c=flow&a=group';
break;
}

```

添加动态流程与添加组织机构类似，都有一个父 ID，从而形成一个树形结构的数据流，与原型图相比，超时的 2 天变成 1 天，对应的值为 86400 秒，单击【增加中转】按钮可再次增加一个子流程，如图 3-12 所示。

图 3-12 添加动态流程页面

添加完流程，就可以通过流程管理模块管理流程了。由于流程支线有时较多，为了防止意外发生，系统将一次性删除整个流程，下一个流程重新添加就可以了。编辑页面在显示时可能稍微复杂一点，这里讲一下流程显示的主要思路。

首先查询数据中所有 `pid=0` 且 `disabled=0` 的主流程，系统会显示这个流程的一些信息，由于很多字段都是数字类型，所以需要通过数组生成字符串，方便用户浏览；然后在流程的结尾判断 `isend` 是否等于 1，如果等于 1 则说明流程结束了且显示【结束】按钮，否则显示【展开子节点】按钮；单击【展开子节点】按钮，前端会通过 Ajax 的方式把 `frompid` 和 `pid` 提交给服务器，通过 `frompid` 来判断是不是同一个流程，通过 `pid` 返回下面的子流程；最后系统以 JSON 格式返回对应的数据组，客户端再次迭代。

这里需要说明的是，为了让客户端和服务端的字段显示一致，字段类型的数组一定要一致。流程编辑页面如图 3-13 所示。



图 3-13 流程编辑页面

客户端代码主要注重数组的内容和对 Ajax 返回子流程的处理,通过 isend 字段判断流程是否结束是一个需要注意的地方,流程编辑页面的代码如下。

```
v/im/flow_edit.html
.....
<div class="cont_right">
    <div class="role_title"> <strong><?php echo $show['name']; ?
></strong> <a href="index.php?m=im&c=index&a=delflow&id=<?php echo
$show['flowid'] ?>">删除整个流程</a></div>
    <div class="Add to cont">
        <style type="text/css">
            /*特殊样式*/
            .flow_table{}
            .flow_table td{border-radius:10px;}
            .flow_table .arrow{border:0px;}
        </style>
        <table class="flow_table">
            <tr>
                <td><?php echo $fromtype[$show['fromtype']];?>【<?php
                    if($show['fromtype']==1) {echo $roleid[$show['fro
mroleid']];}
                    if($show['fromtype']==2) {echo
$moduleid[$show['frommoduleid']];}
                    ?>】</td>
                <td class="arrow">增加一条记录-<?php echo $sendtype[$show
['sendtype']];?></td>
                <td><?php echo $totype[$show['totype']];?>【<?php
                    if($show['totype']==1) {echo
$roleid[$show['toroleid']];}
                    if($show['totype']==4) {echo $moduleid[$show['to
```

```

moduleid']]};}
        ?>】</td>
        <td>通过率<b><?php echo $show['agree']; ?>%</b></td>
        <td>逾期天数为<b><?php echo $show['outdate']; ?></b>天
</td>

        <td>运行状态【<b class="red"><?php echo
$isgood[$show['isgood']]; ?> </b>】</td>
        <td><!--Ajax 判断子节点-->
        <div id="flowid<?php echo $show['flowid']; ?>" >
        <?php if(!$show['isend']){ ?><a href="javascript:;"
onClick="subNode(<?php echo $show['flowid'] ?>,<?php echo
$show['flowid'] ?>)">展开子节点</a><?php ?>
        </div>
        </td>
    </tr>
</table>
</div>
</div>
<script type="text/javascript">
//数组类型
$fromtype = new Array('指定员工','指定角色','指定模块');
$sendtype = new Array('通知','+邮件','+消息','+邮件和消息');
$sendclick= new Array('增加一条记录');
$stotype = new Array('指定员工','指定角色','直属上级','直属下级','
接收模块');
$isgood = new Array('正常','无此角色','无此用户','无直属上级','无
直属下级');

//Ajax 返回子节点
function subNode(pid,flowid){
    //alert(flowid)
    $.post("index.php?m=im&c=index&a=subflow",{pid:pid,flowid
:flowid},function(data){
        //对数组进行处理
        if(data != 0){
            $html = '<table class="flow_table">';
            $data = eval(data);
            for(var i in $data){

```

```

        $html += '<tr><td>→</td><td>' + $data[i].name +
'</td> <td>' + getType($totype,$data[i].totype) + '</td><td>' +
$data[i].toname + '</td>';
        var fid = 'flowid' + $data[i].flowid;
        //判断是否结束
        if($data[i].isend == 1){//结束
            $html += '<td>结束</td></tr>';
        }else{
            $html += '<td><div id="' + fid + '"><a href=
"javascript:;" onClick="subNode(' + $data[i].flowid + ',' + flowid
+ ')">展开子节点</a></div></td></tr>';
        }
    }
    $html += '</table>';
    //alert($html);
    document.getElementById("flowid" +
flowid).innerHTML = $html;
    }else{
        alert(0);
    }
});
}
//返回数组对应的类型
function getType($a,$n){
    return $a[$n];
}
</script>
</body></html>

```

服务端的主要代码是流程列表、删除流程和编辑流程，其中 flowedit 中的数组和客户端的数组要保证一致，代码如下。

```

m/im/index.php
<?php
//流程管理列表
function flowlist()
{
    global $MYSQL;
    //模块列表排序

```

```

        $order = $_GET['order']?$_GET['order']:'addtime';
        if($_SESSION['sort'] == 'DESC'){$_SESSION['sort']='ASC';}else{$_SESSION['sort'] = 'DESC';}
        $sort = $_SESSION['sort'];
        $flow_data = DataList("SELECT * FROM `member_flow` WHERE
`pid` = 0 AND `disabled` = '0' ORDER BY `$order` $sort LIMIT
0,50",$MYSQL);
        include 'v/im/flow_list.html';
    }
    //删除流程, 伪删除
    function delflow(){
        $id = $_GET['id'];
        $name = $_SESSION['uname'];
        global $MYSQL;
        if($MYSQL->query("UPDATE `member_flow` SET `disabled` = '1'
WHERE `frompid` = $id)){
            OALog('',0,$name . '删除流程: '.$id,',','');
            alert('已经删除','index.php?m=im&c=index&a=flowlist');
        }else{
            back('数据库有错误! ',-1);
        }
    }
    // 载入流程编辑模板
    function flowedit(){
        $flowid = $_GET['flowid'];
        global $MYSQL;
        $data = DataList("SELECT * FROM `member_flow` WHERE
`flowid` = $flowid LIMIT 1",$MYSQL);
        $show = $data[0];
        //相关数组
        $fromtype = array(0=>'指定员工',1=>'指定角色',2=>'指定模块');
        $sendtype = array(0=>'通知',1=>'邮件',2=>'消息',3=>'邮件和
消息');
        $sendclick= array(0=>'增加一条记录');
        $totype = array(0=>'指定员工',1=>'指定角色',2=>'直属上级',3=>'直属下级',4=>'接收模块');
        $isgood = array(0=>'正常',1=>'无此角色',2=>'无此用户',3=>'
无直属上级',4=>'无直属下级');
    }

```

```

        //角色数组
        $MYSQL->select_db("ai_admin");
        $role = DataList("SELECT `roleid`,`rolename` FROM
`admin_role`",$MYSQL);
        $roleid = [];
        foreach($role as $v){
            $roleid[$v['roleid']] = $v['rolename'];
        }
        //模块数组
        $module = DataList("SELECT `moduleid`,`module` FROM
`admin_module` WHERE `disabled`=0 ",$MYSQL);
        $moduleid = [];
        foreach($module as $v){
            $moduleid[$v['moduleid']] = $v['module'];
        }
        include 'v/im/flow_edit.html';
    }
    //这里是 Ajax 调用的子流程
    function subflow($pid,$flowid){
        global $MYSQL;
        $pid = $_POST['pid'];
        $flowid = $_POST['flowid'];
        $data = DataList("SELECT * FROM `member_flow` WHERE `pid`
= $pid AND `frompid` = $flowid",$MYSQL);
        //这里最好是一个 JSON 数组
        if($data){
            echo json_encode($data);
        }else{
            echo '0';
        }
    }
    ?>

```

另外，在实际开发中，产品部认为增加一个日历的功能便于添加和显示备忘录和消息公告等，于是在协商后，技术部就找了一个现成的日历插件，产品部觉得不错就同意这个样式了，具体页面如图 3-14 所示。

上一年 上一月 2018年2月▼ 农历戊戌年 【狗年】 下一月 下一年						
星期日	星期一	星期二	星期三	星期四	星期五	星期六
				1 十六	2 世界湿地日	3 十八
4 世界抗癌...	5 二十	6 廿一	7 国际声援...	8 过小年	9 廿四	10 世界气象日
11 廿六	12 廿七	13 廿八	14 腊月二十九	15 除夕	16 春节	17 大年初二
18 大年初三	19 大年初四	20 大年初五 ...	21 大年初六	22 大年初七	23 初八	24 初九
25 初十	26 十一	27 十二	28 十三			

关联选项 ▼ 每天做好一件事！ 日历设置 重要任务 ▼

图 3-14 日历页面

日历不是重点，重点是学习利用现成的插件快速满足产品的要求。很多优秀的程序员做了很多插件和代码，只要复制并按照文档修改一下就好了，修改之前最好看一下文档，如果没有文档就用一些功能强大的编辑器进行查找和替换。比如 Dreamweaver CC，界面左侧是一些格式化的命令按钮，按 Ctrl+F 组合键可以查找和替换，查找和替换还支持正则表达式，功能非常强大，如图 3-15 所示。

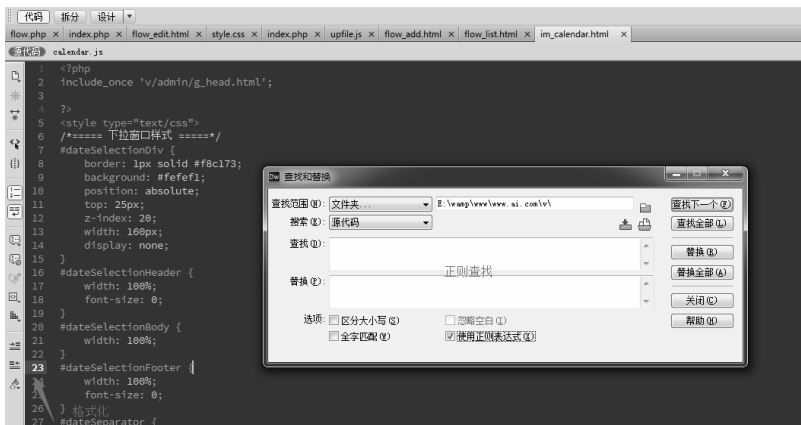


图 3-15 查找和替换

除编辑器外，有些浏览器还自带了一些开发者工具，比如谷歌浏览器中的【更多工具】-【开发者工具】，通过单击、选择就可以看到所要修改

的地方，甚至还有错误提示，非常好用，如图 3-16 所示。另外，火狐浏览器也不错，而且还是中文版的。

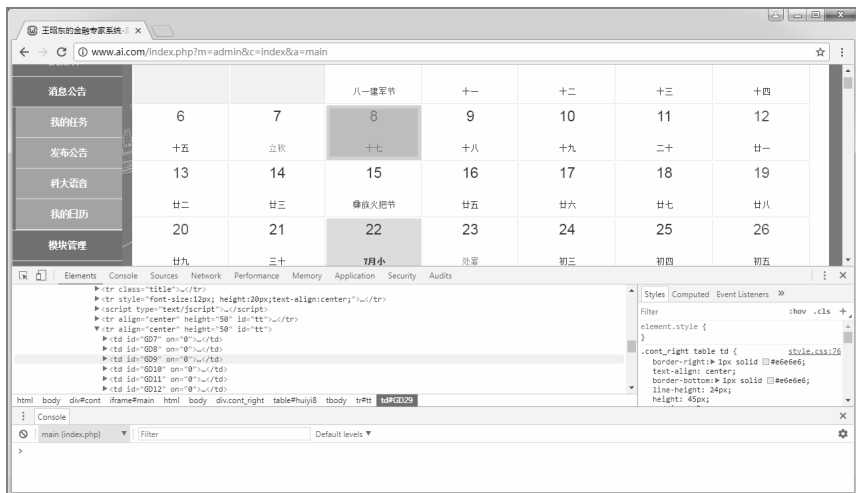


图 3-16 谷歌浏览器中的开发者工具

3.8 系统维护

系统维护默认有一个主界面，这个主界面主要由访问量、磁盘空间、系统日志和系统信息四个部分组成。其中，访问量主要以用户登录为主，由于用户登录都会记录在日志中，所以统计时主要是统计系统日志的登录情况。这里就是一个简单的统计，默认查看本周的登录情况，也就是说最多显示 7 天的登录情况，代码如下。

```
m/ system/ index.php

.....
// 系统维护主界面

function main(){
    global $MYSQL;
    $MYSQL->select_db("ai_log");
    $log_data = DataList("SELECT * FROM `admin_log` ORDER BY
`logid` DESC LIMIT 0,50",$MYSQL);
    $w_arr = array(1=>"周一",2=>"周二",3=>"周三",4=>"周四",
5=>"周五",6=>"周六",7=>"周日");
```

```

//判断当前周
$l1 = date('w');
if($l1 == 0){//周日
    $l1 = 7;
}
//管理员登录日志
$admin_arr = [];
$arr = [];
for($i=1;$i<=$l1;$i++){
    //差值
    $x = $i-$l1;
    $y = $x+1;
    $s_date = date('Y-m-d',strtotime("$x day"));//开始日期
    $e_date = date('Y-m-d',strtotime("$y day"));//结束日期
    $admin_arr[$i] = checking("SELECT `logid` FROM
`admin_log` WHERE `title` = '登录' AND `time` > '$s_date' AND `time`
< '$e_date'", $MYSQL);
    $week[$i] = $w_arr[$i];
}
//会员（含专家）登录日志
$member_arr = [];
for($i=1;$i<=$l1;$i++){
    $x = $i-$l1;
    $y = $x+1;
    $s_date = date('Y-m-d',strtotime("$x day"));//开始日期
    $e_date = date('Y-m-d',strtotime("$y day"));//结束日期
    $member_arr[$i] = checking("SELECT `logid` FROM
`member_log` WHERE `title` = '登录' AND `time` > '$s_date' AND `time`
< '$e_date'", $MYSQL);
}
}
}
.....

```

系统日志就更简单了，主要是一个倒序的查询与显示。系统信息能够调用一些 PHP 系统函数就可以了，比如 `php_uname` 函数，代码如下。

```

m/ system/ index.php
.....
//系统信息

```

```
$sys_os  = php_uname('s');
$sys_v   = php_uname('r');
$sys_php = phpversion();
$sys_cpu = php_uname('m');
.....
?>
```

下面讲一下剩余空间。剩余空间这个功能确实不错，但由于 PHP 是以文件遍历和累加的方式计算一个文件夹空间大小的，所以在性能上可能会有影响，在文件数量较少时可能不明显，但是在海量数据出现时就必须使用第三方私有云工具了，不过仍然可以通过数据库共享的方式定期显示一些磁盘、内存等信息。PHP 剩余空间相关函数的代码如下。

```
m/ system/ index.php
.....
$d_s = disk_free_space('E:');
//已用空间 disk_total_space('E:');
//内存大小 memory_get_usage (true)
$d_m = dirsize('E:\wamp\www\www.ai.com\MP3');
$d_f = dirsize('E:\wamp\www\www.ai.com\files');
$d_t = dirsize('E:\wamp\www\www.ai.com\m');
$d_i = dirsize('E:\wamp\www\www.ai.com\img');

//目录大小，这个很消耗性能，建议使用第三方软件
function dirsize($dir) {
    $dh = opendir($dir);
    $size = 0;
    while ($file = readdir($dh)){
        if ($file != "." and $file != ".."){
            $path = $dir."/".$file;
            if (is_dir($path)){ $size += dirsize($path);} else
if(is_file($path)) { $size += filesize($path);}
        }
    }
    closedir($dh);
    return $size;
}
```

对于磁盘预警，这里由于采用了第三方软件，所以新建了一个空间使

用日志表 (disklist_log), 这个表就属于意外情况下新增的表, 数据库表设计的权限仍然交给 DBA, DBA 设计好后更新数据库设计文档并告知开发人员。为了满足实际需求, 该表的字段可以再多一些, 比如 logid (日志 ID)、url (机房、服务器目录)、type (0 磁盘, 1 内存, 2SSD, 3GPU)、size (可用空间大小)、mp3 (音频、视频)、files (插件)、img (图片)、sys (系统)、bad (坏盘)、time (时间)、israde (伪删除) 和 count (使用率) 等。

异地容灾功能实现起来也比较简单, 其实就是把源码文件等复制到一个远程的服务器目录中, 数据库数据可以选择单独备份。文件恢复就是异地容灾的一个反向工作, 如果对方提供 FTP 等账号密码, 则可以使用 FTP 系列函数, 比如 ftp_connect、ftp_login 和 ftp_put 等 (服务端需要开启 FTP 扩展), 代码如下。

```
m/file/index.php
<?php
.....
function ftp(){
    //传值
    $url = $_GET['url'];
    $user = $_GET['user'];
    $pass = $_GET['pass'];
    $path = $_GET['path'];
    $files= $_POST['files'];//文件列表
    $conn = ftp_connect($url) or die("Could not connect");
    ftp_login($conn,$user,$pass);
    //批量上传
    foreach($files as $file){
        ftp_put($conn,$file,$path.'/'.$file,FTP_BINARY);
    }
    //关闭 FTP
    ftp_close($conn);
    back('FTP 上传成功! ',-1);
}
?>
```

正常情况下我们使用的是第三方管理软件, 仅在第三方管理软件维护后增加一个系统日志。

3.9 任务轮询

常用的任务轮询的方法主要有 4 种：操作系统的定时任务、数据库的触发器、PHP 死循环和服务端软件。其中，服务端软件是最常用的。我们这里主要使用数据库的触发器，比如 MySQL 中的 INSERT 触发器，这项工作在实际工作中有很大一部分交给了 DBA。之所以交给 DBA 主要有两个原因：一是为了安全，减少程序员的数据库操作；二是在实际工作中工作流程不会经常变动。MySQL 触发器（使用 Navicat for MySQL 打开 MySQL）示例如图 3-17 所示。

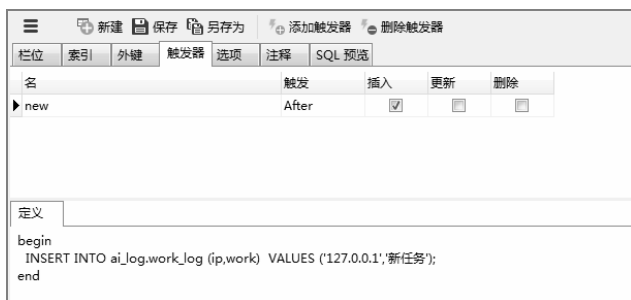


图 3-17 MySQL 触发器

如果项目比较小、公司工作流程比较简单，则可以使用 PHP 死循环的方式进行轮询，这个轮询实际上包含了两个部分，一个是系统级别的轮询，另一个是具体的任务轮询。系统轮询类似一个统一的任务管理，默认管理所有的工作流程，而具体的任务轮询则只管自己的任务，比如某个具体的流程。任务轮询界面如图 3-18 所示。



图 3-18 任务轮询界面

任务轮询的代码可以简单一点，通过超链接的方式打开指定的页面或通过 Ajax 的方式进行请求。Ajax 请求方式的参考代码如下。

```
v/system/working.html
<script type="text/javascript">
//Ajax
function sys(){
    var vie    = $("#ie").val();
    var vsleep = $("#sleep").val();
    var vmax   = $("#max").val();
    //Ajax 请求指定页面，开始工作
    $.get("index.php?m=system&c=work&a=main",{ie:vie,sleep:vsleep,vmax:vmax},function(result){
        alert(result);
    });
}
//Ajax 停止工作
function stop_work(){
    // Ajax 请求指定页面，停止工作
    $.get("index.php?m=ajax&c=index&a=work_txt",{ },function(result){
        alert("已经停止！");
    });
}
</script>
```

系统轮询实际上是一个批量轮询，一次性运行多个任务（流程），每个任务都单独打开一个页面。由于使用 CURL 请求一个页面，1 秒钟后关闭，所以 Web 服务端要开启 CURL，当然也可以使用 `file_get_contents` 等函数。CURL 代码如下。

```
m/system/work.php
<?php
function main(){
    //判断当前用户状态，是否开始、暂停、停止。
    $txt    = '';
    $ie     = $_GET['ie'];
    $sleep  = $_GET['sleep'];
    $vmax   = $_GET['vmax'];
```

```

$work = file_get_contents('server/work.txt');
if($work != ''){
    $txt = '已经开始! ';
    //更改文件
    if(file_put_contents('server/work.txt','')===0){
        $curl = curl_init();
        curl_setopt_array($curl, array(
            CURLOPT_RETURNTRANSFER => 1,
            //网址, 注意不要写错了
            CURLOPT_URL => 'http://' . $_SERVER['HTTP_HOST'] .
            '/server/server.php',
            CURLOPT_POST => 1,
            CURLOPT_TIMEOUT => 1,
            CURLOPT_POSTFIELDS => array('ie' => $ie, 'sleep'
=>$sleep, 'vmax'=>$vmax)));
        curl_exec($curl);
        curl_close($curl);
    }
}
else{
    $txt = '请不要重复开始! ';
}
echo $txt;
}
.....
?>

```

在 `m/system/work.php` 文件中，`main` 方法请求的核心文件是 `server/server.php`，可以说 `server.php` 文件才是任务的大“总管”。`server.php` 通过同目录下的记事本文件 `work.txt` 内容是否大于零来决定任务是否停止。`server.php` 文件代码如下。

```

server/ server.php
<?php
//初始化
ignore_user_abort(TRUE);
$sleep = 1;
$i      = 0;
while(true) {
    //获得 Ajax 提交的参数

```



```

// 是否停止
$work = file_get_contents('work.txt');
//是否关闭浏览器
if(isset($_POST['ie'])) {ignore_user_abort($_POST['ie']);};
//休眠时间
if(isset($_POST['sleep'])) {$sleep = $_POST['sleep'];}
//最长工作时间
if(isset($_POST['vmax'])) {set_time_limit($_POST['vmax']);}
//停止系统并输出到日志
if($work > 0) {
    file_put_contents('log.txt', date('Y-m-d H:i:s'). ',
已经停止' . $i."\\r\\n", FILE_APPEND);
    $txt = '已经停止';
    goto stop;
}
//这里可以使用 foreach 批量打开任务页面, 也可以把具体要执行的任务写
到里面, 比如测试时可以将变量$i 追加到一个日志文件中
    $i++;
    sleep($sleep);
}
//跳出循环
stop:
$i = 0;
?>

```

系统维护还有很多有趣的工作, 比如系统中有一个叫作 **MP3** 的音频文件目录, 这些音频文件实际上是一些客服和销售人员的电话录音, 这时就可以通过 **PHP** 进行简单的声音管理, 代码如下。

```

m/tel/index.php
<?php
//声音列表
function tel()
{
    global $MYSQL;
    //遍历指定目录的文件夹与属性, 目录结构为 MP3/员工 ID/文件名
    $data = [];
    include 'include/mp3file.class.php';//导入 MP3 类
    foreach (glob('MP3/*/*.mp3') as $filename) {

```

```
$a = [];  
$mp3 = iconv('GB2312','UTF-8',$filename);  
$size= round(filesize($mp3)/1024/1024,2) . 'MB';  
$a['path'] = $mp3;  
$a['size'] = $size;  
$a['lasta'] = date('Y-m-d',filemtime($mp3));  
$a['laste'] = date('Y-m-d',filemtime($mp3));  
$a['time'] = mp3Time($mp3);  
preg_match("/MP3\\/\\d+/i",$mp3,$uid);  
$a['uid'] = str_replace('MP3/','',$uid[0]);  
$data[] = $a;  
}  
.....//其他代码，比如获得会员 ID、会员名称等  
include 'v/tel/tel_list.html';  
}
```

3.10 策略管理

量化投资专家系统的知识库是交由专家（实际上是知识工程师）来编写的，为了方便知识库的管理，我们把知识库和推理机结合在一起，最终以交易策略管理的形式展现给用户。对于策略编辑的底层部分，根据具体类别又细分出了函数、决策树、推理和符文 4 个模块组。这些模块组主要具有添加、查看、编辑和删除功能，唯一和普通文章不一样的地方就是添加的内容实际上是一些代码（算法），下面以添加函数模块组为例进行讲解。

添加和编辑函数页面如图 3-19 所示。页面一般由函数样式、函数类别、函数名称、数据接口、PHP 源码和 SQL 语句等主要字段构成。函数样式可分为控件、SQL、插件和接口，默认为“控件”，方便知识工程师选择。函数类别可分为指标、财务、行情、交易、系统、数学和数据，其中“指标”是默认类，方便知识工程师进行区分和选择。数据接口默认只能选择自己添加过的数据接口。PHP 源码是核心，是控件封装的主要内容。SQL 语句只针对 SQL 函数样式起作用，其他的表单主要是辅助作用。

图 3-19 添加和编辑函数页面

添加和编辑函数的核心代码如下。

```
m/expert/index.php
<?php
.....
// 载入函数编辑模板
function editfun(){
    global $MYSQL;
    $funid = $_GET['funid'];
    $dat = DataList("SELECT * FROM `function` WHERE `funid`=
$funid LIMIT 1",$MYSQL);
    $show = $dat[0];

    $MYSQL->select_db("ai_count");
    //返回名下的数据接口
    $uname = $_SESSION['uname'];
    $utype = $_SESSION['utype'];
    $dataid= DataList("SELECT `dataid`,`title` FROM `data_set`
WHERE `username` = '$uname' AND `usertype` = '$utype'", $MYSQL);
    include 'v/expert/function_edit.html';
}
.....
?>
```

函数管理页面是一个列表页，如图 3-20 所示。

函数名称	函数类型	PHP源码	语法说明	使用热度	MD5验证	操作
<input type="checkbox"/> 市盈率	指标	略	市盈率	0	d41d8c-d98-f00b204e9800998ecf8427e	编辑
<input type="checkbox"/> 市净率	指标	略	市净率	0	d41d8c-d98-f00b204e9800998ecf8427e	编辑
<input type="checkbox"/> MA	指标	略	移动均线	0	d41d8c-d98-f00b204e9800998ecf8427e	编辑
<input type="checkbox"/> MACD	指标	略	指数平滑移动均线	0	d41d8c-d98-f00b204e9800998ecf8427e	编辑

图 3-20 函数管理页面

函数管理服务端代码如下。

```
m/expert/index.php
<?php
.....
//函数管理列表
function functionlists()
{
    global $MYSQL;
    //分页
    $data = DataList("SELECT * FROM `function` LIMIT 0,100",$MYSQL);
    include 'v/expert/function_list.html';
}
.....
?>
```

上面讲解了函数的实现思想和过程，其实决策树、推理和符文的实现思想和过程也是一样的，之所以分成不同的模块组主要是考虑到未来权限的划分，让不同的知识工程师添加不同的知识，所有的知识其实都有一个 PHP 源码编辑的部分，满足产品要求的源码高亮显示，这里主要使用 PHP 的自带函数 `htmlspecialchars_decode`，并且通过 Ajax 的方式进行简单调用，当然，如果开发组有更多的时间，也可以使用第三方已经写好 Web 编译器。PHP 高亮显示的主要代码如下。

```
m/ajax/highlight.php
<?php
// PHP 源码高亮
// 载入主桌面
function main()
{
    //br 转回车符
```

```

    $php = trim($_POST['code']);
    $php = str_replace(array("\r\n", "\r", "\n"), "", $php);
    $php = strip_tags($php, '<br>,<div>');
    $php = htmlspecialchars_decode($php);
    $php = str_replace('&nbsp;', ' ', $php);
    $php = str_replace('</div>', '', $php);
    $php = str_replace(array('<br>
/>', '<br>', '<div>'), "\r\n", $php);
    $php = highlight_string($php, true);
    $php = str_replace(array("\r\n", "\r", "\n"), "", $php);
    //file_put_contents('tt.txt', $php);
    die($php);
}
?>

```

另外，产品要求增加一个知识库导入的功能，方便把其他专家系统的知识导入到本地数据库，这时可以使用 SQL 文件，也可以使用 CSV 文件，使用 CSV 文件可以更好地进行修改，比较符合实际情况。为了方便上传 CSV 文件，我们在客户端使用了一个叫作 upfile.js 的文件，这是一个简单的 JavaScript 上传控件，代码如下。

```

v/expert/fact_add.html
.....
<input type="button" value="添加 CSV 文件"

onClick="iframe_upload('index.php?m=data&c=upload&a=up_file&call=
addFile')" />
    <script type="text/javascript"
src="static/upfile.js"></script>
    <script type="text/javascript">
//上传文件后处理结果
function addFile(str){
    document.getElementById("csvfile").value = str;
}
</script>
.....

```

服务端使用已经定义好的 csv_get_lines 方法读取 CSV 文件，代码如下。

```
include/upload.php
```

```
// $lines 表示读取的行数, $offset 表示开始位置
function csv_get_lines($csvfile, $lines, $offset = 0) {
    if(!$fp = fopen($csvfile, 'r')) {
        return false;
    }
    $i = $j = 0;
    while (false != ($line = fgets($fp))) {
        if($i++ < $offset) {
            continue;
        }
        break;
    }
    $data = array();
    while(($j++ < $lines) && !feof($fp)) {
        $data[] = fgetcsv($fp);
    }
    fclose($fp);
    return $data;
}
```

服务端逐行读取 CSV 文件并存入到指定的 fact 表中, CSV 文件格式如图 3-21 所示 (范文为 files/fact.csv)。

	A	B	C	D
1	reg	pids	description	ratio
2		112	知识库	0

图 3-21 fact.csv 格式

服务端具体的操作代码如下。

```
m/ expert/index.php
.....
//开始导入到知识表
function inputcsv()
{
    global $MYSQL;
    //导入 CSV 文件的读取方法
    include_once 'include/upload.php';
    $csv = $_POST['csvfile'];
```

```

$data = csv_get_lines($csv,2,1); //从第2行开始
//添加到数据库
$ok = 0;
$username = $_SESSION['uname'];
foreach($data as $v){
    //格式
    $sql = "INSERT INTO `fact`(`reg`,`pids`,`description`,`ratio`,`username`) VALUES('$v[0]','$v[1]','$v[2]','$v[3]','$username')";
    if($MYSQL->query($sql)){
        $ok++;
    }else{
        echo $sql;
        die;
    };
}
$txt = '成功导入' . $ok . '条';
alert($txt,'index.php?m=expert&c=index&a=factlist');
}

// 使用 MD5 验证文件是否存在
function md5php(){
    global $MYSQL;
    $md5 = md5(html2php($_POST['code']));
    $utype = $_SESSION['utype'];
    $uname = $_SESSION['uname'];
    $data = DataList("SELECT `funname`,`username` FROM `function` WHERE `md5` = '$md5' AND `username` <> '$uname' AND `utype` <> '$utype' LIMIT 1",$MYSQL);
    if($data){
        echo json_encode($data);
    }else{
        echo '0';
    }
}

//HTML 标签转换为 PHP 文本
function html2php($post){
    $php = trim($post);

```

```
$php = str_replace(array("\r\n", "\r", "\n"), "", $php);
$php = strip_tags($php, '<br>,<div>');
$php = htmlspecialchars_decode($php);
$php = str_replace('&nbsp;', ' ', $php);
$php = str_replace('</div>', '', $php);
$php = str_replace(array('<br
/>', '<br>', '<div>'), "\r\n", $php);
return $php;
}
?>
```

对于策略的管理，虽然第一期可视化编辑的功能不多，但还是尽量做得比较可用一些。

在实际工作中，第一期有程序员协助交易员编写代码，包括很多自定义的函数、决策树，而符文这个功能几乎不怎么用。

下面实现一个 Ajax 的小功能，如图 3-22 所示是 Web 编辑器。



图 3-22 Web 编辑器

客户端代码如下。

```
m/expert/game_add_menu.html
<hr />
<ul class="title">
  <li class="l1"><?php echo $edit==0?' hover':' l2';?>><span
onClick="menu_select('index.php?m=expert&c=index&a=addgame&edit=
```



```

0')">控件模式</span></li>
    <li class="l1"<?php echo $edit==1?' hover':' l2';?>"><span
onClick="menu_select ('index.php?m=expert&c=index&a=addgame&edit=
1')">符文模式</span></li>
    <li class="l1"<?php echo $edit==2?' hover':' l2';?>"><span
onClick="menu_select('index.php?m=expert&c=index&a=addgame&edit=
2')">源码模式</span></li>
    <li class="l1"><span onClick="submenuAJAX('fun')">函数指标
</span></li>
    <li class="l1"><span onClick="submenuAJAX('tree')">算法决策
</span></li>
    <li class="l1"><span onClick="alert('开发中')">调试预览
</span></li>
    <li class="l1"><span onClick="alert('开发中')">帮助
</span></li>
</ul>
<script type="text/javascript">
//添加用户到表单中
function submenuAJAX(s){
    switch(s){
        case 'fun':
            $url = "index.php?m=ajax&c=index&a=selectfun";
            break;
        case 'tree':
            $url = "index.php?m=ajax&c=index&a=selecttree";
            break;
    }
    //Ajax 返回用户名列表
    $.post($url,{type:0},function(data){
        if(data != '0'){
            subSelect(data,'phpeidt',2,1);
        }
    });
}
</script>
<script type="text/javascript" src="static/upfile.js"></script>
//服务端查询示例, 返回前3个
m/ajax/index.php

```

```
//返回决策树列表
function selecttree()
{
    //权限验证模块
    $type = $_POST['type'];
    global $MYSQL;
    $MYSQL->select_db("ai_tree");
    $data = DataList("SELECT `tree`.`treeid`,`tree`.`title`,`tree_data`.`content` FROM `tree` LEFT JOIN `tree_data` ON `tree`.`treeid` = `tree_data`.`treeid` LIMIT 0,100",$MYSQL);
    if($data){
        echo json_encode($data);
    }else{
        echo '0';
    }
}
```

表格多选框其实是一个插件，这个插件可以是一个浮动的窗口，也可以是一个面板，样式部分交给美工就好，下面主要讲解客户端返回的 JSON 数组是如何处理的。首先通过 eval 将 JSON 字符串转换成对象，然后通过 for in 进行遍历，遍历时注意截取对应的 KEY。一般都是通过.name 的方式获得对应的 JSON 值，但是为了通用性，建议使用数字作为下标返回其对应的值，所以这时可以使用自定义的 getKey（对象、序号）返回对应的值。getKey 其实就是把对象重新遍历成数组，然后通过序号返回对应的值。最终生成 HTML 字符串，通过 innerHTML 方式追加到指定的 HTML 标签中，示例代码如下。

```
static/upfile.js
//多选框，这个通用性不强，应该提取出共性来
subSelect = function(data,ss,n1,n2){
    // 获得鼠标指针位置
    e = window.event;
    xy = mouseCoords(e);
    div = document.createElement("div");
    div.style.position = "absolute";
    div.style.top = xy.y + 'px';
    div.style.left = xy.x + 'px';
```

```

        document.body.appendChild(div);
        var $html = '<div class="selectTable"><span onclick="self_
close(this);">关闭</span><br />';

        //遍历 JOSN, 追加表单
        $data = eval(data);
        for(var i in $data){
            $html += '<span onClick="SelectPHP(\' ' +
getKey($data[i],n1) + '\',\'' + ss + '\')">' + getKey($data[i],n2)
+ '</span>';
        }
        $html += '</div>';
        div.innerHTML = $html;
    }
    function getKey(O,n){
        var tt = new Array();
        $i = 0;
        for (var p in O){
            tt[$i] = O[p];
            $i++;
        }
        return tt[n];
    }
    //被选中的表单
    SelectPHP = function(n,ss){
        document.getElementById(ss).innerHTML += n;
    }

```

3.11 数据统计

3.11.1 数据接口

数据接口主要是为了对接第三方数据和大型基金公司的自有数据，数据默认以 SQL 语句为主，当然也可以使用 CSV 或 JSON。数据接口的主要

功能是数据或与接口连接相关的表单项的增加和修改，核心字段是地址、数据库名、用户名、密码和 Token（令牌）。

3.11.2 数据查询

为了方便用户查询，允许用户对已有的数据接口直接编写 SQL 语句，而对于不会编写 SQL 语句的用户，则提供一个可视化的高级搜索界面，通过选择下拉菜单和复选框自动生成对应的 SQL 语句，数据查询（高级搜索）界面如图 3-23 所示。

加入菜单接口类型数据视图易加选项请输入关键字搜索

添加接口aljm请输入接口关键字搜索

指定数据库

SELECT * FROM `member_group` WHERE `groupid` > '0'

SQL区

flow_logmember_addressmember_flowmember_groupmember_memorandummessagemessage_datamessage_set对应表

显示	排序	改名	字段	类型	说明	运算符	数值
<input type="checkbox"/>	0	groupid	groupid	int(10) unsigned	职位ID	>	0并且
<input type="checkbox"/>	0	groupname	groupname	varchar(25)	职位名称	=	并且
<input type="checkbox"/>	0	mainusername	mainusername	varchar(50)	负责人姓名	=	并且
<input type="checkbox"/>	0	time	time	int(10) unsigned	添加时间	=	并且
<input type="checkbox"/>	0	text	text	char(80)	岗位简介职责	=	并且
<input type="checkbox"/>	0	members	members	varchar(500)	下属会员集合, 分隔	=	并且
<input type="checkbox"/>	0	pids	pids	varchar(25)	上级ID集合, 分隔	=	并且
<input type="checkbox"/>	0	disabled	disabled	tinyint(1) unsigned	是否删除, 0正常, 1待删除	=	并且

警告050运行SQL加入SQL操作

图 3-23 数据查询界面

这种 Web 形式的数据查询代码可能比较多，核心代码及解释如下。

(1) 遍历自己的数据接口，代码如下。

```
m/count/index.php
function main()
{
    $menu = 'data';
    //搜索对应的表
```

```

$q = $_GET['q']?$_GET['q']: '';
global $MYSQL;
$username = $_SESSION['uname'];
$type = $_SESSION['utype'];
$sql = "SELECT `dataid`,`title` FROM `data_set` WHERE
`username` = '$uname' AND `usertype` = '$utype' AND `title` LIKE
'%" . $q . "%' LIMIT 0,25";
$dataid = DataList($sql,$MYSQL);
include 'v/count/index.html';
}

```

(2) 前端生成下拉菜单，代码如下。

```

v/count/index.html
.....
<li>
    <select name="execut_type" class="add_select"
style="width: 100px;" onChange="select_ database(this.value)">
        <option value="0">数据接口</option>
        <?php foreach($dataid as $v){ ?>
            <option value="">?php echo
$v['dataid']; ?>><?php echo $v['title']; ?></option>
        <?php } ?>
    </select>
</li>
.....

```

(3) 选择对应的数据接口，返回其所有表，并以表格的形式显示到前端，同时对不同的表单添加不同的名称，方便客户端获取对应表单的值，代码如下。

```

v/count/index.html
//表名
istable = '';
function select_table(dataid,t){
    istable = t;
    //用 Ajax 方式返回表的字段
    $.get("index.php?m=ajax&c=index&a=tables",{dataid:dataid,
table:t},function(data){
        $data = eval(data);
        $html = '<table cellpadding="0" class="staff_title">

```

```
<tr><th>显示</th><th>排序</th><th>改名</th><th>字段</th><th>类型</th><th>说明</th><th>运算符</th><th>数值</th></tr>';
    for(var i in $data){
        $html += '<tr><td> <input type="checkbox"
name="checkboxs" /> </td><td><input name="sorts" value="0"
style="width:50px;" type="text" /></td><td><input name="names"
value="' + $data[i].COLUMN_NAME + '" type="text" /></td><td
name="colname">' + $data[i].COLUMN_NAME + '</td><td>' +
$data[i].COLUMN_TYPE + '</td><td>' + $data[i].COLUMN_COMMENT +
'</td>';

        $html += '<td><select name="operator"><option
value="">= </option><option value="">></option><option
value="">=></option><option value=""><</option><option
value="">=<=</option><option value="">!=>!= </option><option
value="LIKE">LIKE</option><option value="NOT LIKE">NOT
LIKE</option><option value="IN">IN</option></select></td>';

        $html += '<td><input name="numbers" type="text"
style="width:75px;" /><select name="myOperator"><option
value="AND">并且</option><option value="OR">或者
</option></select></td></tr>';
    }
    $html += '</table>';
    document.getElementById("tables").innerHTML = $html;

});
}
```

返回指定数据接口中的所有表，主要由 3 个步骤构成。首先通过数据接口的 ID 返回对应的数据接口配置数组；然后利用数据接口配置数组连接对应的数据库，如果连接成功，则通过“SELECT 'table_name' FROM 'tables' WHERE 'table_schema' = '数据库名' ”返回对应的表，再通过“SELECT 'COLUMN_NAME','COLUMN_COMMENT','COLUMN_TYPE' FROM 'columns' WHERE 'table_schema' = '数据库名' AND 'table_name' = '表名' ”返回指定表的字段；最后以 JSON 形式返回数组，代码如下。

```
m/ajax/index.php
//返回指定数据接口中的所有表
function tables(){
```

```

        $dataid = $_GET['dataid'];
        $set = [];
        //通过$dataid 查询对应的数据接口并赋值给$set 数组, 方便连接数据接口
        $CONN = new mysqli($set['ip'],$set['user'],$set['password']);
        $CONN->set_charset('utf8');
        $CONN->select_db($set['database']);
        if($CONN->connect_errno){
            //连接失败
            echo 0;
        }else{
            $CONN->select_db("information_schema");
            $database = $set['database'];
            $sql = "SELECT `table_name` FROM `tables` WHERE `table_schema` = '$database' LIMIT 0,50";
            $tables = DataList($sql,$CONN);
            //判断是否要返回表的字段
            $table = $_GET['table'];
            if($table){
                //这条 SQL 语句就是重点
                $sql = "SELECT `COLUMN_NAME`,`COLUMN_COMMENT`,`COLUMN_TYPE` FROM `columns` WHERE `table_schema` = '$database' AND `table_name` = '$table' LIMIT 0,30";
                $data = DataList($sql,$CONN);
                if($data){
                    $_SESSION['dataid'] = $dataid;
                    $_SESSION['table'] = $table;
                    echo json_encode($data);
                }else{
                    echo '0';
                }
            }else{
                //返回列表
                echo json_encode($tables);
            }
        }
    }
}

```

(4) 代码加入到 SQL 文本框主要是通过 document.getElementsByName 获得表单数组, 然后遍历对应的表单值, 代码如下。

```

v/count/index.html
//加入到 SQL 文本框
add_sql = function(){
    //遍历表单
    // 要把表名和字段连起来，这样兼容性强
    $txt = 'SELECT * FROM \'' + istable + '\'';
    $numbers = document.getElementsByName("numbers");
    $operator = document.getElementsByName("operator");
    $colname = document.getElementsByName("colname");
    $names = document.getElementsByName("names");
    $sorts = document.getElementsByName("sorts");
    $myOperator= document.getElementsByName("myOperator");
    $checkboxxs = document.getElementsByName("checkboxxs");
    // 指定元素
    for(var i=0;i<$numbers.length;i++){
        if($numbers[i].value != ''){
            $txt += ' WHERE \'' + $colname[i].innerHTML + '\''
+ $operator[i].value + ' \'' + $numbers[i].value + '\'' +
$myOperator[i].value;
            // 去掉最后几个字符
            $txt = $txt.substr(0, $txt.length - 3);
        }
    }
    //是否显示字段
    $select = ' ';
    $a_sort = new Array();
    $s = 0;
    for(var s=0;s<$checkboxxs.length;s++){
        //判断是否选中
        if($checkboxxs[s].checked){
            //对数组排序
            $vv = new Array();
            $vv[0] = $sorts[s].value;
            $vv[1] = s;
            $a_sort[$s] = $vv;
            $s++;
        }else{
    }
}

```



```

    }
    //判断排序，由大到小
    $n_sort = $a_sort.sort(compare);
    //组合数组元素
    for(var vv=0;vv<$n_sort.length;vv++){
        var s = $n_sort[vv][1];
        //判断是否改名，如果不相等就修改
        var $name = $names[s].value;
        var $col = $colname[s].innerHTML;
        if($col != $name && $name != ''){
            $select += '`' + $col + '` AS `' + $name + ``,`';
        }else{
            $select += '`' + $col + ``,`';
        }
    }
    //去掉最后一个字符
    $select = $select.substr(0, $select.length - 1);
    //替换*号
    if($select){
        $txt = $txt.replace(/\*/i, $select);
    }else{
        alert($select);
    }
    $("#mysql").val($txt);
}

```

(5) 运行 SQL 语句，默认使用 cookie 的方式保存 7 天，方便用户下次调用，然后通过 cookie 开始查询，代码如下。

```

v/count/index.html
//运行 SQL 语句
run_sql = function(){
    //默认保存 7 天
    $.cookie("DATAID",mydataid,{expires:7});
    $.cookie("LASTSQL",$("#mysql").val(),{expires:7});
    $.cookie("LIMIT",$("#limitid").val(),{expires:7});
    $.cookie("LENGTH",$("#lengthid").val(),{expires:7});
    menu_select('index.php?m=count&c=index&a=view&menu=view');
}

```

```
m/count/ index.php
// view方法就是跳转到对应的视图页面中
function view(){
    $menu = 'view';
    $submenu= '2d';
    include 'v/count/view.html';
}
```

3.11.3 统计视图

根据产品的要求，视图可以有很多种，最简单的就是一个二维表格。二维表格实际上是一个渲染的过程，通过\$.cookie方法返回对应的数据接口ID、SQL和查询数量，然后通过\$.post方法以Ajax方式返回对应的数据，剩下就是渲染工作了。二维表格非常简单，只要遍历对应的<tr><td>内容</td></tr>行就可以了，代码如下。

```
v/count/view.html
<script type="text/javascript">
//20 档标签，多一行数据
//二维图表
post_sql = function(){
    // 开始数据请求
    dataid = $.cookie("DATAID");
    sql    = $.cookie("LASTSQL");
    limit  = $.cookie("LIMIT");
    length = $.cookie("LENGTH");
    $.post("index.php?m=count&c=index&a=AJAXDatabase",{dataid:
:dataid,sql:sql,limit:limit,length:length},function(result){
        //alert(result);
        if(result != 0){
            woking_2d(eval(result));
        }else{
            //停止任务
            alert('你还没有选择相应的数据接口! ');
        }
    })
}
```

```

    });

}
post_sql();

//生成二维表格
working_2d = function($arr){
    //清空原有数据
    $("#tables").html("");
    //这里注意键值对、字段名称和字段值
    //优先处理表头
    $th = $arr[0];
    var dw = 1020;
    var dh = 600;
    if($("#data_w").val()){dw = $("#data_w").val();}
    if($("#data_h").val()){dh = $("#data_h").val();}
    $html = '<table cellspacing="0" class="staff_title"
style="width:'+ dw +'px; height:'+ dh +'px;"><tr>';
    for(i in $th){
        $html += '<th>' + i + '</th>';
    }
    $html += '</tr>';
    for(var i=0;i<$arr.length;i++){
        $html += '<tr>';
        for(ii in $arr[i]){
            $html += '<td>' + $arr[i][ii] + '</td>';
        }
        $html += '</tr>';
    }
    $html += '</table>';
    document.getElementById("tables").innerHTML = $html;
}
</script>

```

\$post 请求的 AJAXDatabase 方法比较简单, 连接数据库并查询, 然后通过数据接口 ID 和对应的 SQL LIMIT 进行查询, 最终以 JSON 方式返回。客户端显示效果如图 3-24 所示。



图 3-24 客户端显示效果

与二维表格类似，为了方便用户以更多的角度更加清楚地展示数据，还可以利用第三方插件显示数据，这里只讲 echarts.js 中的折线图，前面的代码和二维表格的代码一样，先调用 cookie，然后通过 AJAXDatabase 方法返回数据。这里简单讲一下折线图渲染，其他统计图的制作大同小异，代码如下。

```
v/count/view_line.html
.....
working_line = function($arr){
    //清空原有数据
    $("#tables").html("");
    $th = $arr[0];
    var $data = new Array();
    $i = 0;
    for(i in $th){
        $data[$i] = i;
        $i++;
    }
    var $title = new Array();
    for(var i=0;i<$arr.length;i++){
        $title[i] = '记录'+i;
    }
    //开始循环
    $value = new Array();
    for(var i=0;i<$arr.length;i++){
        //生成记录
        var $values = new Array();
        $ii = 0;
        for(ii in $arr[i]){
```

```

        var $number = $arr[i][ii] * 1;
        $values[$ii] = $number;
        $ii++;
    }
    $value[i] = {name:$title[i],type:'line',stack: '总量
',data:$values};
    }
    option = {
        title: {
            text: '折线图'
        },
        tooltip: {
            trigger: 'axis'
        },
        legend: {
            data:$title
        },
        grid: {
            left: '3%',
            right: '4%',
            bottom: '3%',
            containLabel: true
        },
        toolbox: {
            feature: {
                saveAsImage: {}
            }
        },
        xAxis: {
            type: 'category',
            boundaryGap: false,
            data: $data
        },
        yAxis: {
            type: 'value'
        },
        series: $value
    };

```

```

var myid = 't' + 0;
var dw = 1020;
var dh = 700;
if($("#data_w").val()){dw = $("#data_w").val();}
if($("#data_h").val()){dh = $("#data_h").val();}
$("#tables").append('<div id="' + myid + '" style="width:'+
dw +'px; height:'+ dh +'px;"></div>');
var dom = document.getElementById(myid);
var myChart = echarts.init(dom);
myChart.setOption(option, true);
}

```

关于数据统计部分，简单地介绍一下日志系统。这个日志最好是一个通用的函数，在所有有关数据库增加、删除和修改的地方添加这个函数。这个函数至少要有用户名、操作类型和时间 3 个字段，也可以根据自己的需要来设置日志。由于所有的系统都可能用到这个函数，所以建议把这个函数放到 `config.php` 文件中，代码如下。

```

base/config.php
.....
// 添加记录、表名、类型、操作标题、请求的内容、返回的内容
function OALog($table,$type=0,$title,$querystring,$da)
{
    global $data;
    $CONN = new mysqli($data['hostname'],$data['username'],
$data['password']);
    $CONN->set_charset($data['charset']);
    $CONN->select_db("ai_log");
    $utype = $_SESSION['utype'];
    $uid    = $_SESSION['uid'];
    $uname  = $_SESSION['uname'];
    $ip     = $_SERVER['REMOTE_ADDR'];
    $time   = date('Y-m-d h:i:s',time());
    //用户类型为管理员
    if($utype === 0){
        //日志类型，1 为导入，2 为导出，3 为备份，4 为错误，5 为警告，6 为
数据库错误
        $sql = "INSERT INTO `admin_log`(`type`,`title`,

```

```

`querystring`, `data`, `adminid`, `adminname`, `ip`, `time`)
        VALUES ('$type', '$title', '$querys
tring', '$da', '$uid', '$uname', '$ip', '$time')";
    }else{
        switch($table){
            case 'member_log':
                $sql = "INSERT INTO `member_log`(`type`, `title`,
`query string`, `data`, `memberid`, `membername`, `ip`, `time`)VALUES
('$type', '$title', '$querys tring', '$da', '$uid', ' $uname', '$ip', '
$time')";

                break;
            case 'log':

                break;
            default:
                $sql = "INSERT INTO `member_log`(`type`, `title`,
`querys tring`, `data`, `memberid`, `membername`, `ip`, `time`)VALUES
('$type', '$title', '$querys tring', '$da', '$uid', ' $uname', '$ip',
'$time')";

        }

    }
    if($CONN->query($sql)){
    }else{
        echo $sql;
        die;
    }
}
?>

```

日志的使用方法是哪里需要在哪里添加，比如在用户登录操作中，登录成功后日志的代码如下。

```

m/user/login.php
.....
//追加日志
OALog('',0,'登录','', '');
.....

```

3.12 网络爬虫

数据可以不断地积累，这是一个长期的工程，虽然短期不会有明显的效果，但是随着时间和数据的增加，自有数据的规模效应会更加明显。

数据的来源多种多样，网络爬虫就是其中一种获取数据的方法，尤其是定向爬取。量化投资专家系统第一期要实现的就是一个简单的定向爬虫，第二期会通过单独的爬虫服务器集群对海量的非结构化数据进行全自动结构化查询。

首先新建一个爬虫数据库 `ai_spider`，然后在该数据库中创建一个爬虫任务表 `robots`，如表 3-1 所示。

表 3-1 爬虫任务表

字段	类型	长度	默认	注释	主键
id	smallint	5	0	网址 ID	*自增长，主键
url	char	255	Null	网址	*
root	varchar	50	Null	采集根目录	
frequency	varchar	100	0	关键字频率	
title	char	50	Null	网站名称	
description	char	255	Null	网站说明	
type	tinyint	2	0	文章分类	
username	varchar	20	Null	用户名	
listreg	varchar	200	Null	列表正则	
titlereg	varchar	200	Null	标题正则	
keywordsreg	varchar	200	Null	关键字正则	
descriptionreg	varchar	200	Null	描述正则	
bodyreg	varchar	500	Null	内容的正则替换	
charset	char	10	Null	编码类型	
level	tinyint	11	1	级别，采集深度	
starttime	int	11	0	开始时间	
endtime	int	11	0	结束时间	
lasttime	int	11	0	最后采集时间	
disabled	tinyint	1	0	是否启用，0 为启用，1 为禁用	

对应的网址编辑页面如图 3-25 所示。为了方便管理员添加采集的正则，一般会自动填充一些常用的正则，比如关键字正则、标题正则等。

存储格式 默认

网站名称 法律法规

网站描述 最新颁布国家法律法规

URL地址

根目录

关键字频 0

列表正则 /(.)/i

采集层级 1

标题正则 /<h2>(.)</h2>/i

关键字正则 /<meta name="keywords" content="(.)" \>/i

描述正则 /<meta name="description" content="(.)" \>/i

内容正则 /<div class="content">([s|S]*)</div class="dede_pages">/i

编码类型

是否启用 ☐ 禁用 ☒ 启用

保存

图 3-25 网址编辑页面

文章采集的网址一次可能有多个，所以可以用一个批处理文件来遍历这些需要采集的网址。采集的函数可以使用 `file_get_contents` 或 `curl_init`，想并行采集可以使用 `curl_multi_init` 函数，代码如下。

```
v/data/spider_work.php
<?php
set_time_limit(60000);
//设置表头，方便展示进度
$txt .= '<table width="100%" border="1" align="center"
cellspacing="0" cellpadding="0">';
$txt .= '<caption>文章采集测试</caption>';
$txt .= '<tr><th>序号</th><th>类型</th><th>标题</th><th>关键字
</th><th>描述</th><th>内容</th></tr>';
// 输出表头
echo $txt;
ob_flush();
flush();

// 获得需要采集的 ID 集合，最多 30 条
$id = $_GET['id'];
```

```

if($id){
    $ids = array();
    $ids[] = $id;
}else{
    $ids = $_POST['ids'];
}
$ids = implode(',',$ids);
$sql = 'SELECT * FROM `robots` WHERE `id` IN (' . $ids . ') LIMIT
0 , 30';
$data = DataList($sql,$MYSQL);
if(!count($data)){
    goto stop;
}

// 开始遍历数据库并逐条采集
foreach($data as $v){//首先是列表
    $url =$v['url'];
    $list = curl_init();
    curl_setopt($list, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($list, CURLOPT_HEADER, 0);
    $useragent = $_SERVER['HTTP_USER_AGENT'];
    curl_setopt($list, CURLOPT_USERAGENT,$useragent);
    curl_setopt($list, CURLOPT_URL, $url);
    $listhtml = curl_exec($list);
    curl_close($list);
    $utf = strpos($listhtml,'utf-8');
    //判断 meta 头文件中是否包含“utf-8”字样
    if($utf===false){$listhtml =
iconv("gbk","utf-8",$listhtml);}
    preg_match_all($v['listreg'],$listhtml, $urls);
    $urls = $urls[$v['number']];
    //print_r($urls);
    //die();
    // 开始执行
    $i = 1;
    foreach($urls as $r){//然后是正文
        // 查找网址并替换根目录
        $url = strpos($r,'http://')?$r:$v['root'] . $r;
        // 注意 HTTPS, 此处需要开启 open_ssl
    }
}

```

```

        if(checking('SELECT `id` FROM `log` WHERE `url`=\''.$url .'\'' LIMIT 1')){
            continue;
        };
        $ch = curl_init();
        curl_setopt($ch, CURLOPT_URL, $url);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
        curl_setopt($ch, CURLOPT_HEADER, 0);
        curl_setopt($ch, CURLOPT_USERAGENT,$useragent);
        $html = curl_exec($ch);
        curl_close($ch);
        // 这个地方不严谨, 因为还有“utf-8”
        $pos = strpos($html,'utf-8');
        if($pos===false){$html = iconv("gbk","utf-8",$html);}
        preg_match($v['titlereg'],$html, $title);
        $title = $title[1];
        $title = strip_tags($title);
        $title = trim($title);
        preg_match($v['keywordsreg'],$html, $keywords);
        $keywords = $keywords[1];
        $keywords = strip_tags($keywords);
        $keywords = trim($keywords);
        preg_match($v['descriptionreg'],$html, $description);
        $description = $description[1];
        $description = strip_tags($description);
        $description = trim($description);
        // 截取第一个
        $description = preg_replace("/。.*\/i", "",
$description);
        preg_match($v['bodyreg'],$html, $body);
        $body = $body[1];
        $body = preg_replace("</script.*>.*<\/script>/isU", "",
$body);
        /* 特殊处理部分
        // JavaScript 广告等过滤, 也可以后期处理
        $body = preg_replace("</div class=tit twffan=\"done\">
推荐阅读<\/div>(.*?)<P>/isU", "", $body);
        */

```

```

        $body = preg_replace("/<p class=\"sherry_labels\"
style=\"margin:0 ;> (.*)</p>/isU", "", $body);
        $body = strip_tags($body, '<p><br /><br>');
        // 判断是否包含 p 标签
        if(strpos($body, "<p>")>1){}else{
            // 在句号后面添加两个换行符
            $body = str_replace(".", ".<br /><br />", $body);
        };
        $time = time();
        if($title != '' && $body != ''){
            //此处根据需要生成 HTML 页面或 PDF、PNG 等
            tohtml($url,$name,$toname);
            echo '<tr><td>' . $i . '</td><td>' .
            $arr[$v['type']]. '</td><td>' . $title . '</td><td>' .
            $keywords . '</td><td>' . $description . '</td>';
            //这里是静态文件地址
            echo '<td><a href="http://news.meifenqian.com/show.
php?id=' . $lastid . '" target="_blank">浏览</a></td></tr>';
            $i++;
        }else{
            echo '<tr><td colspan="5">标题或者文章为空</td><td><a
href="" . $url . '" target="_blank">' . $url . '</a></td></tr>';
        }
        ob_flush();
        flush();
    }
}
stop:
//输出表尾
echo '<tr><td colspan="6">采集已经停止</td></tr></table>';
?>

```

上面的代码生成 HTML 静态页，非常简单，难点是把 HTML 页面生成 PDF，这就需要借助第三方的类库了，比如 `html2pdf`。生成 HTML 函数的代码如下。

```

// 生成静态页
function tohtml($url,$path,$topath){
    $a = file($url);

```

```

    $f = fopen($path, 'w');
    foreach($a as $k => $v){
        $str = $v;
        fwrite($f,$str);
    }
    fclose($f);
}
//PDF 生成 PNG 图片代码
function pdf2png($PDF,$Path){
    if(!extension_loaded('imagick')){
        return false;
    }
    if(!file_exists($PDF)){
        return false;
    }
    $IM = new imagick();
    $IM->setResolution(120,120);
    $IM->setCompressionQuality(100);
    $IM->readImage($PDF);
    foreach ($IM as $Key => $Var){
        $Var->setImageFormat('png');
        $Filename = $Path.'/'.$md5($Key.time()).'.png';
        if($Var->writeImage($Filename) == true){
            $Return[] = $Filename;
        }
    }
    return $Return;
}

```

html2pdf 的参考网址是 <https://github.com/lester/html2pdf>。

3.13 图片管理

机器学习尤其是深度学习，需要大量的图片数据。比如，用计算机看很多股票 K 线形态，再根据次日的涨跌幅作为标签，让计算机进行监督学习，这样学习了大量的图片之后，就可以利用其生成的模型进行未来趋势

判断。为了生成这些图片，可以按照如下步骤进行操作。

(1) 添加股票行情数据接口。

(2) 生成任务列表。

(3) 设计 K 线前端页面，然后将图片转成 Base64 字符编码，交给 Ajax 进行提交。

(4) 服务端按照目录要求生成图片。

数据接口部分在 3.11 节中已经介绍过，现在讲解任务列表。这个任务列表主要是为了并发考虑的，比如将 2000 多只股票分成 6 个进程(伪进程)，这样每个进程就只有 300 多只股票了，实际上可以通过“生成图片”模块添加很多新的进程，生成的任务界面如图 3-26 所示。

图 3-26 添加图片生成任务界面

在图 3-26 中，K 线类型、SQL 接口 ID 和 SQL 语句这三个选项是非常重要的。K 线类型决定调用哪个模板，SQL 接口 ID 决定连接哪个股票行情数据库，SQL 语句决定模板中所有数据的格式。

在设计 K 线前端页面也就是模板时，主要调用百度 echarts.js。以黑色 K 线形态模板为例，模板函数代码可能如下。

```
server/k/black.php
<?php
```

```
//关闭浏览器也能用
ignore_user_abort(TRUE);
//永不超时
set_time_limit(0);
error_reporting(0);
//获得相关数据
$id = $_GET['id']?$_GET['id']:0;
.....
?>
```

为了初始化,可以根据 ID 获得对应的数据,同时加载 echarts.js 和 jquery.min.js,它们可以被一起使用,代码如下。

```
<!doctype html>
.....
<!--加载 js-->
<script type="text/javascript" src="/static/ echarts.js
"></script>
<script type="text/javascript" src="/static/jquery.min.js">
</script>
</head>

<body>
<!--设置 K 线图像生成区域-->
<div id="container" style="width:<?php echo $width; ?>px;height:
<?php echo $height; ?>px;"></div>
<div id="test"></div>
<input type="button" onClick="save_image()" value="保存" />
```

根据 echarts.js 提供的开发文档,发现 echarts.js 可以通过 Base64 字符编码进行图片文件传输,echarts.js 内置了一个 myChart.jsgetDataURL(),其中 myChart 就是绘图区域的对象,代码如下。

```
<script type="text/javascript">
//初始化相关数据
var sql = "<?php echo $sql; ?>";
// 返回函数
var imgname= 0;
var dom = document.getElementById("container");
var myChart = echarts.init(dom);
```

```
// Base64 位上传图片
save_image = function(){
    picBase64Info = myChart.getDataURL();
    dom.get;
    //通过 xmlhttp64 位上传
    $.post("post.php",{data:picBase64Info,path:path,maxfile:ma
axfile,imgname:imgname},function(result){
        //成功后,开始上传下一个K线形态
        if(result){
            document.getElementById("test").innerHTML =
result;

            post_sql();
        }
    });
}
```

对于客户端轮询,一方面用\$.post 提交之后的返回值是否正确来决定是否查询和生成下一张 K 线图片,另一方面用 window.setTimeout 定时器定时把图片推送到服务器,window.setTimeout 这个 JavaScript 方法的具体时间需要根据经验来确定,时间短了可能图片上传不完整,时间长了性能又不好,可以根据浏览器和本地电脑的性能来设置,比如设置为每秒五张图片,代码如下。

```
//Ajax 从数据库中获取数据
post_sql = function(){
    // 开始数据请求
    $.post("sql.php",{dataid:dataid,sql:sql,workid:workid,slep:slep,fun:fun,down:down,up:up},function(result){
        //alert(result);
        if(result != 0){
            document.getElementById("test").innerHTML =
result;

            woking(eval(result)); //调用K线生成方法
            //图片生成时间间隔
            timer = window.setTimeout("save_image()",200);
        }else{
            //停止任务,注意入口文件的路径
            $.get("../../index.php?m=ajax&c=index&a=updateworking",{n
```



```

:1,i:workid,sql:sql},function(result){
    if(result == 1){
        //跳转到 test.php
        window.location.href = '../../test.php?id
= ' + workid;

    }else if(result == 0){
        alert('停止任务');
    }else{
        document.getElementById("test").innerHTML
= result;
    }
});
}
});
}
post_sql();

```

读者可能会觉得生成 K 线图片是一个难点，实际上恰恰相反，网络上已经有人为我们提供了各式各样的 JavaScript 插件和示例代码，只要稍微用点心，将代码“拿”过来稍加改动就可以了。比如，echarts.js 就非常好用，只要在数组返回时设置成需要的字符串格式就行了。echarts.js 的使用代码大致如下。

```

//开始绘制
function woking(rawData){
    var app = {};
    option = null;
    app.title = '';
    var dates = rawData.map(function (item) {
        return item[0];
    });
    imgname = dates[0];
    var data = rawData.map(function (item) {
        return [+item[1], +item[2], +item[3], +item[4]];
    });
    option = {
        .....
        //参考百度 echarts.js 示例代码
    }
}

```

```

.....
};
if (option && typeof option === "object") {
    myChart.setOption(option, true);
}
}
</script>
</body>
</html>

```

服务端的处理比较简单，利用 `base64_decode` 函数生成图片，然后把结构追加到数据库的 `img_log` 表中，方便客户端查看进度，代码如下。

```

server/k/post.php
<?php
//设置编码格式
header('Content-type:text/html;charset=utf-8');
//读取图片文件，转换成 Base64 编码格式
$base64_image_content = $_POST['data'];
$path                  = $_POST['path'];
$maxfile               = $_POST['maxfile'];
$imgname               = $_POST['imgname'];
//保存 Base64 字符串为图片，匹配图片的格式
if (preg_match('/^(data:\s*image\/(\w+);base64,)/', $base64_
image_content, $result)){
    $type = $result[2];
    // 判断目录是否存在，若不存在，则创建目录
    if(!is_dir($path)){
        //注意操作系统的默认编码
        mkdir(iconv("UTF-8", "GBK",$path),0777,true);
    }
    //判断目录中的文件个数，这个暂不处理，留给分布式存储，方便日后迁移
    //$glob = count(glob($path."/*.*"));
    //$md5   = md5($base64_image_content);
    $new_file = $path . '/' . $imgname . ".{$type}";
    if (file_put_contents($new_file, base64_decode(str_replace(
$result[1], '', $base64_image_content)))){
        //生成记录
        $MYSQL->select_db("ai_log");

```

```

        $uname = $_SESSION['uname'];
        $path0 = 'server/k/' . $new_file;
        $time = date('Y-m-d H:i:s');
        $sql = "INSERT INTO
`img_log`(`username`,`md5`,`path`,`time`)
        VALUES ('$uname','$md5','$path0','$time')";
        $MYSQL->query($sql);
        echo $new_file;
    }else{
        echo '';
    }
}
?>

```

模板页面的代码中有一个 POST 方式请求的 `sql.php` 文件，这个文件的主要作用就是根据数据接口连接指定数据库，然后利用给定的 SQL 语句返回符合模板格式的数据。核心代码如下。

```

server/k/post.php
<?php
    session_start();    // 开启 SESSION，利用 SESSION 保存记录，也可以用
文件判断 SESSION 是否存在
    //
    .....
    //数据接口
    if(!$_SESSION[$sid]){
        include '../base/config.php';
        $MYSQL->select_db("ai_count");
        //查找数据接口配置
        $data = DataList("SELECT `set` FROM `data_set` WHERE `dataid`
= '$dataid' AND `disabled` = 0 LIMIT 1",$MYSQL);
        .....
        //连接数据接口
        $CONN = new
mysqli($set['ip'],$set['user'],$set['password']);
        $CONN->set_charset('utf8');
        $CONN->select_db($set['database']);
        if($CONN->connect_errno){
            //连接失败

```

```

        $_SESSION[$sid] = 0;
    }else{
        $_SESSION[$sid] = MyDataList($sql,$CONN);
    }
}
.....

```

如果包含返回当前记录、处理步长等程序，则重新保存 SESSION；如果 SESSION 不存在，则查询数据库，处理 SQL、SESSION 数组，这里可以根据需要直接把数据进行分组，比如按涨跌幅分组，代码如下。

```

.....
//这里需要返回符合前端要求的数据格式
function MyDataList($s,$mysql){
    $q = $mysql->query($s);
    $data = array();
    if($q){
        while($row = mysqli_fetch_assoc($q)) {
            $data[] = array_values($row);
        }
    }else{
        $data = 0;
    }
    return $data;
}
?>

```

为了方便对图片生成的子进程进行管理，在图片列表模块中添加了【生成】、【编辑】、【运行】和【停止】按钮。单击【编辑】按钮可以编辑具体的生成图片选项。单击【运行】按钮，Ajax 通知服务端根据生成图片（子进程）的相关设置复制或删除指定的 K 线模板文件，服务端代码如下。

```

m/ajax/ index.php
.....
//更新生成图片的工作状态
function working(){
    $n    = $_GET['n'];
    $i    = $_GET['i'];//进程 ID
    $k    = $_GET['k'];

```

```

$username= $_SESSION['uname'];
//这里要生成一个文件，用于网页并发
if($n == 1){
    //文件名称
    $file    = 'server/k/' . $k . '.php';
    //新文件名称，注意新文件名中间有进程的ID号
    $newfile = 'server/k/' . $k . '_' . $i . '.php';
    copy($file, $newfile);
}else if($n == 0){
    // 删除文件
    unlink('server/k/' . $k . '_' . $i . '.php');
}
global $MYSQL;
$MYSQL->select_db("ai_tree");
//更新进程状态
if($MYSQL->query("UPDATE `image_work` SET `woking` = '$n'
WHERE `id` = $i AND `username`= '$username' LIMIT 1")){echo
'1';}else{echo '0';};
}
.....

```

如果批量处理很多股票，则可以建立一个记事本文件，用来逐行存储对应的股票代码，顺序执行完一只股票的操作后就删除一行代码，以此类推，直至结束，更新生成图片的工作状态，代码如下。

```

//更新生成图片的工作状态
function updateworking(){
    $i    = $_GET['i'];
    $sql  = $_GET['sql'];
    $username= $_SESSION['uname'];
    global $MYSQL;
    $MYSQL->select_db("ai_tree");
    //返回第一行并删除
    $txt      = 'server/k/' . $i . '_tables.txt';
    $tables   = file($txt);
    $tablename = array_shift($tables);
    $tablename = str_replace(array("\r\n", "\r", "\n"), "",
$tablename);
    //使用分隔符连接并保存

```

```

        $newtxt = implode('',$tables);
        $f = file_put_contents($txt,$newtxt);
        //替换表名
        $sql = preg_replace('/(FROM)\s+(\`)\w+(\`)/i','FROM ' .
"$tablename`",$sql);
        //判断是否需要转义
        $sql = str_replace("'",'"',$sql);
        $path = 'img/k/' . $i . '/' . $tablename;
        if($tablename){
            if($MYSQL->query("UPDATE `image_work` SET `sql` =
'$sql`,`name` = '$tablename`,`path` = '$path`,`working` = '1' WHERE
`id` = $i AND `username` = '$uname' LIMIT 1")){
                echo '1';
            }else{
                echo $MYSQL->error;
            }
        }else{
            echo '0';
        }
    }
    ?>

```

对应的前端 JavaScript 代码如下。

```

v/image/image_list.html
.....
<script type="text/javascript">
work = function(n,i,k){
    $.get("index.php?m=ajax&c=index&a=working",{n:n,i:i,k:k},
function(result){
    if(result == 1){
        alert('更新成功! ');
        //刷新当前页面
        window.location.reload();
    }
    });
}
</script>

```

图片列表界面如图 3-27 所示。

图片名称	记录条数	通知方式	操作
hist_sh600000_m5	1	295392870@qq.com	生成 编辑 停止
hist_sh600000_m5	1	295392870@qq.com	生成 编辑 停止
hist_sh600000_m5	1	295392870@qq.com	生成 编辑 停止
hist_sh600000_m5	1	295392870@qq.com	生成 编辑 停止

图 3-27 图片列表界面

在图片列表界面中单击【生成】按钮，系统将把任务提交给服务端中转文件 server/k/work.php?id=29，其中的“29”是具体的 ID 号，主要代码如下。

```
server/k/work.php
<!doctype xmlns="http://www.w3.org/1999/xhtml">
<html>
<head>
<meta charset="utf-8">
<title>生成图片</title>
</head>

<body>

<?php
//做一个连接，生成图片
$id = $_GET['id'];
include 'base/config.php';
$MYSQL->select_db("ai_tree");
$username = $_SESSION['uname'];
if($username && $id){
    //指定任务 ID
    $data = DataList("SELECT * FROM `image_work` WHERE `username`
= '$username' AND `working` = 1 AND `id` = $id LIMIT 1", $MYSQL);
    $show = $data[0];
    ?>
    <a href="<?php echo $show['k']. '_'. $show['id']; ?>.php?sql=
<?php echo $show['sql']; ?>&id=<?php echo $show['id']; ?>&type=
<?php echo $show['type']; ?>&width=<?php echo $show['width']; ?>
&height=<?php echo $show['height']; ?>&path=<?php echo $show
['path']; ?>&maxfile=<?php echo $show['maxfile']; ?>&slep=<?php
echo $show['slep']; ?>&dataid=<?php echo $show['dataid']; ?>"
id="myhref">开始生成图片</a>
```

```

<?php
}else{
    echo 'no login';
}
?>

<script type="text/javascript">
//这里是一个小的任务轮询，用于连接不同的进程页面
window.setTimeout(function(){
    document.getElementById("myhref").click();
},1000);
</script>
</body>
</html>
    
```

在实际的项目中，是分成很多 Web 服务器进行图片生成的，每个 Web 服务器只负责不到 10 个进程，这样就能够较短的时间（一个月）内生成数以千万张的图片。在图片生成时，如果需要浏览器打开 `ignore_user_abort(false)` 才能工作，那么当前浏览器的性能要明显高于失去焦点的浏览器的性能。具体的实现流程和方法会在项目拓展中进行介绍。

另外，如果小图片非常多，多到几百万张，那么在没有专业软件的情况下可以尝试使用存储模式添加压缩文件到共享磁盘中，这样比很多小文件一起传快一些，如图 3-28 所示。

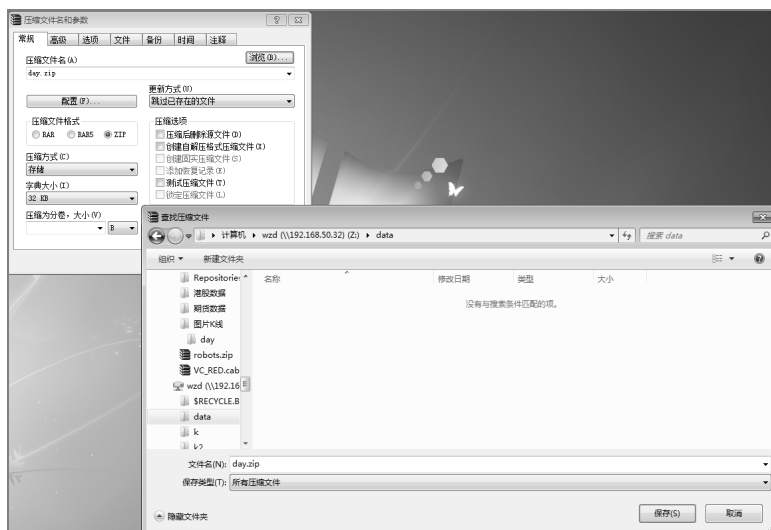


图 3-28 小图片压缩后进行批量传输

3.14 机器学习

机器学习虽然是第二期的项目，但是在第一期我们就可以把它嵌入到系统中，方便知识工程师快速了解机器学习。不过，由于开发人员、知识工程师与机器学习工程师尤其是深度学习工程师是三拨不同的人，所以真实的生产环境是深度学习工程师仍然使用自己的深度学习框架，比如使用 TensorFlow 进行深度学习，然后根据深度学习得出的模型生成真实的条件单。

为了解决这个问题，一方面需要加强普通开发人员的机器学习能力，另一方面要利用 PHP-ML 等进行改造，使其不仅能够进行图片深度学习，也能够进行数值数据方面的深度学习（尤其是金融领域的深度学习框架）。

PHP-DL 在第二期中会有一些测试，在第一期中主要使用交换文件的方式，自动读取深度学习框架生成的条件单，然后自动化交易（模拟交易）。下面简单讲解 PHP-ML 的安装过程，方便 PHP 开发人员继续学习机器学习和深度学习。

（1）PHP-ML 需要 PHP 7.0 以上版本的支持，为了安装方便也可选择集成安装环境，比如 WampServer 3.0.6（64 位）或 phpStudy 2016.11.03。

（2）Composer 的安装方法可以参考 <https://getcomposer.org/download/>，比如在 Windows 版本下直接下载安装 composer-setup.exe 64 位即可。

（3）安装 Composer 后就可以到 GitHub 上下载 PHP-ML 的镜像了，其中一个源码，另一个是示例 git 命令：

```
git clone https://github.com/php-ai/php-ml.git
git clone https://github.com/php-ai/php-ml-examples.git
```

（4）利用 CMD 命令进入到项目目录下，然后安装 Composer Install 文件，稍等几分钟，如果项目目录中出现 vendor 文件夹就表示安装成功了（这个环节是最核心的），如图 3-29 所示。

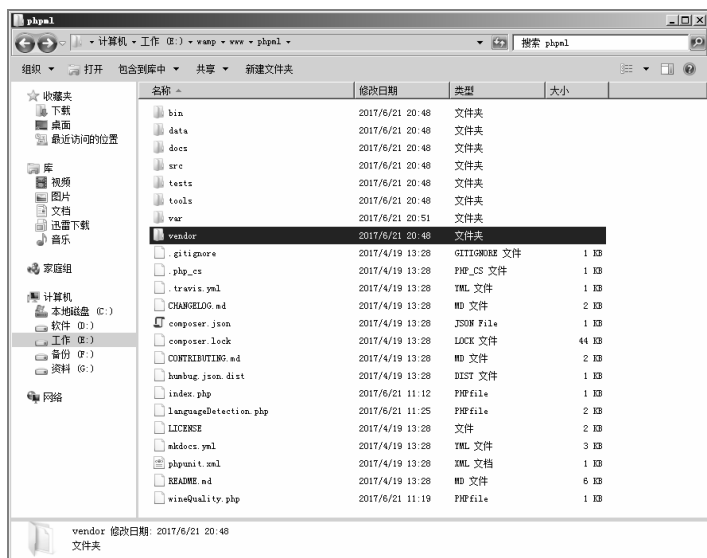


图 3-29 Composer 安装成功

(5) 复制 php-ml-examples 目录中的示例，测试结果如图 3-30 所示。

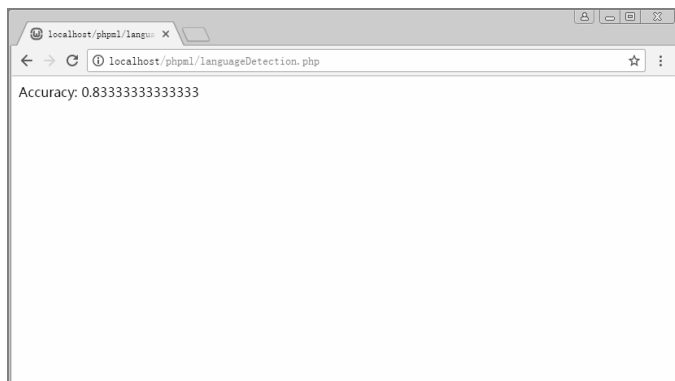


图 3-30 PHP-ML 测试页面

如果觉得 PHP-ML 太麻烦，也可以利用数学方法进行分类，比如方差、中位数、线性回归等。如果觉得 TensorFlow 不错，也可以使用 TensorFlow 作为编程语言的一个接口。除此之外，还有很多优秀的深度学习和机器学习框架，比如 PYTorch（以 Python 语言为主）、Synaptic.js（以 JavaScript 为主）、CNTK（以 C#为主）和 Deeplearning4j（以 Java 为主）。

考虑到项目的完整性，我们举一个机器学习的例子，这个机器学习的主要作用就是寻找规律和优化参数。虽然我们倾向于成立一个独立的机器学习小组，把生成好的策略直接推送到策略库中，但是实际上我们还是希望开发人员有一定的机器学习基础，机器学习开发的部分也可以自己来编写，不过这需要开发人员拥有一定的机器学习算法基础。

3.15 验收报告

关于测试工具，这里主要是指黑盒测试，具体测试工具如 TestMaker 等这里不做过多介绍，只简单讲一下验收报告。一般来说，验收报告是针对外包项目的，这里可以把技术部看做是一个外包公司，把产品部看做是甲方，验收报告一般都是由甲方出具的，不过在验收前，技术部会出具一个《使用说明书》，这个《使用说明书》未来还可以申请软件著作权。《使用说明书》比较简单，包括系统、运行环境、功能截图（截图时尽量截取完整）和说明。《使用说明书》最简单的制作方法就是通过产品原型图直接生成。下面是验收报告的大概格式。

量化投资专家系统验收报告

一、项目基本信息

项目名称：量化投资专家系统

项目合同甲方：产品部

项目合同乙方：技术部

合同类型：技术开发合同

二、验收目的

目的在于对项目进行全方位的测评，检验技术部提供的系统是否符合

合（这里特指产品原型）产品部的要求。

如果是项目外包，那么就要根据合同（主要是需求说明）进行验收了。

三、验收范围

产品部按照产品原型进行项目验收，并着重第一期项目的验收，对于系统的扩展部分也要做一些适当的了解，对于模块化的功能要逐一验收其是否运行正常。

四、项目验收表

内容	工具	问题	建议	备注
夏普率	无	计算公式错误	$sharpeRatio = \frac{E(R_p) - R_f}{\sigma p}$	参考夏普比率

验收人：产品部

在实际开发中，项目很少能够一次性通过，在验收时会出现很多问题，这也是要全面测试的原因。全面测试的好处是问题出现后可以一次性修改，项目验收人员不能因为怕麻烦就不去详细检查，也不能因为个人审美问题而增加功能以外的样式和布局。

问题出现以后，要增加新功能是需要另议的。新功能不能包含在测试报告中，即使这个功能看起来非常简单也不可以。

最后再说一下测试，项目在移交前会在产品部进行内部测试，在内部测试之前有一个单元测试。所谓的单元测试，在本系统中主要是指模块测试。绝大部分问题都会出现在单元测试和内部测试中。

一般项目的整个测试周期和开发周期都是相对很长的。不过在测试时技术开发的工作量的确少了不少，只要主要的工程师跟进项目测试就可以了。

4

第 4 章

经典策略解析

在阅读本章内容之前，读者最好拥有股票交易的相关操作经验，这样才能进一步系统地了解量化投资专家系统是怎样构建的，也更加容易弄清很多经典策略背后的意义。本章没有任何源码，非常适合那些对策略感兴趣的读者。由于本书不是交易员指南，所以大部分都是经典的交易策略。

通过本章的学习，可以比较概括地知道一些策略的构成要素和长时间实现良好收益的基本原因。了解这些知识也方便程序员进行系统开发和策略编写。当然，就本系统而言，策略编写好后并不表示可以上线了，因为作为一套企业级的量化投资专家系统，策略的上线还需要走一定的流程。策略上线的基本流程如图 4-1 所示。

一般都说策略重要，那么如何衡量一个策略的好坏呢？有没有一个统一标准？也许有人会说一定要看夏普率、超额收益率、盈亏比和杠杆比例，有些人甚至罗列得更加细致，但是大部分投资人主要还是看投资经理的风格，而一般老百姓则只关注投资的这笔钱到期能够产生多少回报。投资人一般也不管投资经理具体买了哪种类型的股票或做了什么对冲，他们只看重投资经理投资的风格和最终到手的收益是多少。

我们知道，一般基金的收益（利润）会随着资金体量的增加、收益率的提高和时间的延长而成体积式地增加，如图 4-2 所示。

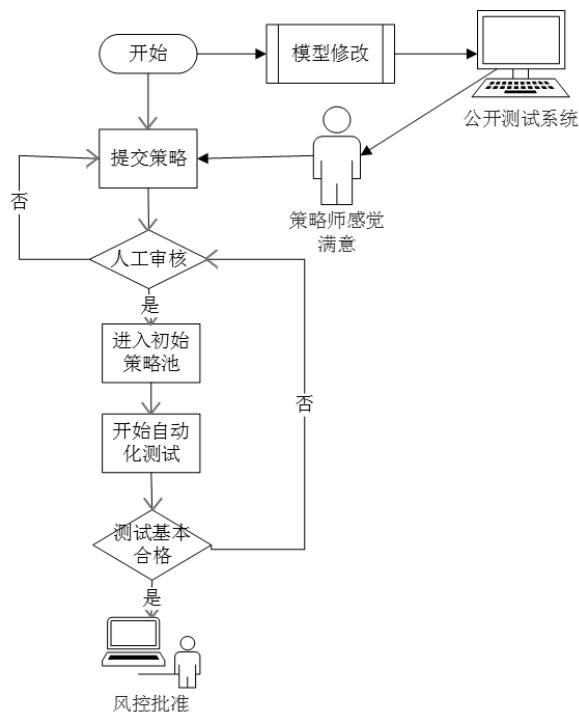


图 4-1 策略上线的基本流程

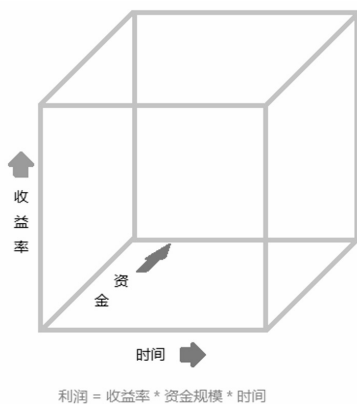


图 4-2 回报率三要素

具体是哪些因素影响了利润呢？单从交易员的角度来看，这个因素可能多得无以计数。就拿股票来说，能够影响股价变动的因素可能有宏观因素、财务因素、技术因素、时间因素和交易因素，而每一个因素都可能包

含更多的子因素。

比如在宏观因素方面，可能就需要考虑政治的稳定性、国民生产指数、国民消费指数、通胀指数、银行利率、税收政策、就业率、金融所占经济比重、人口老龄化程度，甚至是国际安全和美股指数。如果把宏观缩小为一个行业，那么就需要考虑这个行业是否是国有企业、是否是朝阳产业、是否是刚需、是否是垄断行业，以及行业规模和所属资本派系等。如果具体到某一只股票，那么我们可能更加关心股票企业的经营数据，比如收益、负债、销售，以及工商、法务、公告甚至是创始人的个人信息。有些更细致的投资公司甚至还会关注这家企业自身所处的企业生命周期在哪里（是成长型企业，还是成熟型企业亦或是衰老型企业），与之对标的企业是谁，企业的相对估值是多少，企业的有形资产、无形资产、净资产比例分别是多少，正常利润扣除后稳定收入来源是哪项，企业客户群是什么性质的，以及投资后是否会产生协同效应等。

如果这些都需要量化，就会试图通过输入较小的变量来寻找那些能够引起较大变化的量，也就是所谓的投资切入点。

影响股票价格变动的因素数不胜数，有些因素彼此独立，而有些因素就像蝴蝶效应一样相互影响，常见的因素如图 4-3 所示。

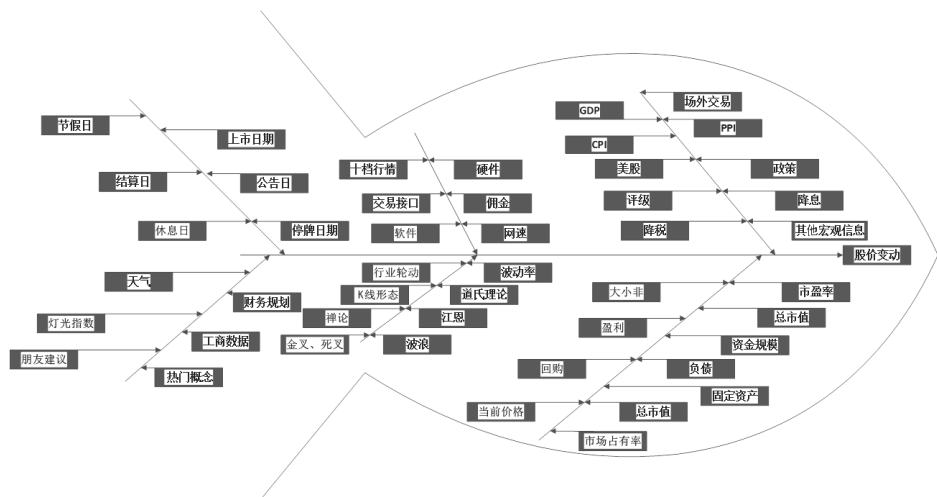


图 4-3 影响股价变动的因素

除上面密密麻麻的、看着都头疼因素外，读者可能还会根据自己的经验、偏好而分析很多别的因素，比如著名的裙子理论、建筑物周期理论、执政党执政周期理论、随机森林理论等。每一个大类都由若干个小类构成，如果愿意的话找出几万个是没有问题的，不过实际中哪些更好就需要验证了，有时因为工作的不同所找到的因素也会不同，比如宏观分析分员和 IT 人员考虑的问题几乎完全不相关，但是好像都有道理。

1. 资金管理

谁都想第一个发现那些能够造成高壁垒、高利润、好管理、定价能力强、有成长空间、发展稳定的因素，但是事实上你永远不是第一个发现某种因素的人，也不是唯一的一个。普通股民可能并不关注这些，因为大部分人都是投机的。而如果去问一个资金管理人员，他可能就会从另一个方面说“什么因素都不如资金管理重要，人不是神，谁也不知道未来会怎样，所以做好资金管理工作就是最好的策略”。如果你问他什么是资金管理，他也许会说：“资金是生命，而资金管理就是解决如何保护生命的。”说的不无道理，在投资活动中，仓位控制与资金管理非常重要，尤其是当资金规模较大且资金来源又较多时。

资金管理一般是通过仓位控制来进行风险管控的。比如，通过单笔交易亏损比例的风险百分比或止损幅度来计算仓位大小。

最常见的资金管理就是单笔交易亏损比的风险百分比。比如，风控比例为 5%，总资金为 100 万元，则允许的最大亏损为 5 万元（即 $100 \text{ 万元} \times 5\%$ ），如果产生亏损，则剩余资金的 5% 作为下一笔交易的最大亏损即 $(100 - 5) \times 5\% = 475 \text{ 万元}$ 。

当然，在两次亏损之间要进行反思，汲取经验后再交易（或者把亏损比例由 5% 降到 2%），否则长此以往钱会亏没的。确定了单笔交易允许的亏损金额后，就可以计算仓位的大小了。

另一种算法是累计最大亏损比，比如 25%，允许连续亏损 5 次，那么每次的最大亏损比就是 5%。这个 5 次一般要和持仓周期关联，默认是 1 天，

5次就是一周，所以一周之内就要平仓。

当然，也可以用亏损比例进行资金管理。我们都知道，亏损 20%需要盈利 25%才能持平，亏得越多涨回来的百分比越大，如果亏损 50%，涨回来就需要 100%。一只股票涨 100%是比较难的，但是跌 50%却是有可能的，所以亏损比例即便很大也不要超过 35%。

最后希望投资者能够留下 3%的救命钱，无论行情多么得好，总要留下 3%的救命钱，甚至当资金较少时可以留下 5%的保命钱。即便想满仓进入，还是要考虑一下破产风险，否则一旦发生意外怎么办？

如果非要找一个公式来进行量化管理，则首推凯利公式。凯利公式（凯利方程）非常有用，除可将长期增长率最大化外，它绝不允许在交易中失去全部现有资金。凯利公式非常简捷：

$$f^* = p - (1 - p) / (rW/rL)$$

式中， f^* 为现有资金应进行下次投注的比例； p 为获胜率； rW 是获胜后的净盈率， rL 是净损失率， rW/rL 是盈亏比。

如果是专家，更喜欢靠经验进行规划，那么一般来讲降低杠杆、控制好头寸就没有多大问题了，当然这样风险小了挣钱也少了。要想挣得多，风险也就越高。多空双方两者整体收益（所有交易者的收益总和）而言是相当的，即：高风险 × 高收益 = 低风险 × 低收益。

2. 分步建仓

下面说一下建仓的问题。最常用的方法就是排队，投资者感觉这个价位合理就买入，或者直接挂在那里等着；还有就是条件单，在规定的时间内，当价格等满足一定的条件就买入。但是以上两种方法对资金规模较大的基金就不适用了，那么大一笔资金不可能以某一个固定的价位成交，这时就需要一些建仓的手段，最简单的是拖拉机下单。拖拉机下单就是对于某一个价位区间，平均隔几分钟就买入一定量的股票。还有一种方法是金字塔下单方式，与之对应的就是倒金字塔下单方式，网上还有网格下单方式。金字塔下单方式很简单，就是盈利时不断加仓，当行情反转时开始陆

续减仓，如图 4-4 所示。

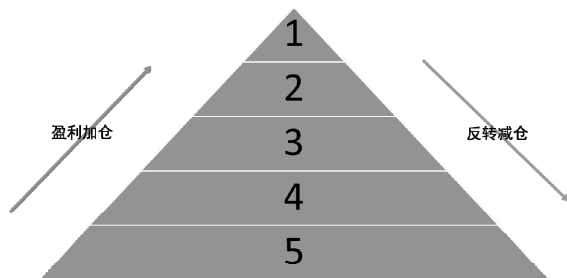


图 4-4 金字塔建仓法

3. 止损

一个完整的策略一般是要有止损的，没有只涨不跌的股票，任何股票都有泡沫，任何公司都有其生命周期。曾经的“石油大王”被“钢铁大王”取代，曾经的“超市大王”被“软件大王”取代，谁敢保证现在的“互联网大王”不被人工智能或其他的“大王”取代呢？

关于交易需要注意的事项就这么多，具体的执行还是仁者见仁、智者见智。虽然操作手法彼此不同，但是大体上我们还是知道那些偏向宏观的人更喜欢做大资金的长远投资，而偏向微观的人更喜欢通过数学模型来做高频交易。

我们把这些不同的交易周期整合到系统里，相信它很快就变成一个非常综合的系统了，如图 4-5 所示。



图 4-5 交易的周期

虽然实际上很难有一个基金公司同时具备这么多周期的策略，但是笔者还是按照完整的周期策略进行讲解，毕竟有很多短期投机者最后都变成了长期持有者。

4. 操作思路

下面简单讲解基本操作思路。

操作的前提是：市场允许交易，并且交易的周期足够长，最好有一定的波动率，比如港股。

（1）要了解宏观数据

这里的宏观具体数据有哪些不重要，关键对于市场而言，投资者的心里要有一个大致的预判，其实预测的结果无非就是以下三种：

- 认为价格较低；
- 认为价格较高；
- 观望与不知所措。

上面每一种预测结果都是有道理的，所以要相信自己的判断。因为市场不论是上涨、下跌还是震荡，都有它的道理。有些人天生就适合做多，比如沃伦·巴菲特，而有些人就比较适合做空，比如乔治·索罗斯，还有些人就喜欢那些不确定的行情，比如詹姆斯·西蒙斯。巴菲特、索罗斯和西蒙斯三位著名的交易员如图 4-6 所示。



图 4-6 巴菲特、索罗斯、西蒙斯

如果你是巴菲特，那么就该在你认为价格较低时买入，因为你找到了那些被人低估的资产。

如果你是索罗斯，那么就该在你认为价格较高时卖出，因为你找到了那些被人高估的泡沫。

如果你是西蒙斯，那么就该在你认为价格不明时买卖，因为你找到了那些被人忽视的概率。

当然，以上三位也并不都是单一地做多、做空或波段，我们只是从他们的战绩上看他们更擅长某一方向而已。

（2）选择熟悉的行业

人的生命有限，经历不同，所以对同一个事物的理解也不一样，有些人小时候衣食无忧，而有些人则颠沛流离；有些人对数学感兴趣，而有些人则对科学情有独钟。正是因为这些客观的事实造成的差异，才形成了他们今天的投资偏好。

如果你不了解这个行业，就很难取得好成绩，就算你是股神巴菲特，也很可能在科技股票里栽跟头。因为你不懂科技，不懂互联网，你怎么知道这只股票价格是高了还是低了呢？

不仅行业如此，二级市场也是一样，比如巴菲特青睐股市，索罗斯青睐外汇，西蒙斯青睐期货，本质上也是和游戏规则暗合的。

选择一个行业最简单的方法就是回忆一下自己的人生经历，看看自己对哪些行业更熟悉，如果实在没有也没有关系，只要选择喜欢的行业就好了，毕竟兴趣才是最好的老师，不要担心兴趣帮不了你，因为喜欢找游戏规则漏洞的索罗斯和喜欢数学的西蒙斯一样都是可以盈利的，只要关键时学会那么一点点的触类旁通，捅破那层所谓的窗户纸就可以了。

至于市场的游戏规则，多看几遍、多试几次就好了，相信很快就能找到喜欢的领域。

最后告诉读者一个小窍门，那就是想办法爱上你选择的行业市场并对它负责，爱上它可能有这样几个理由：

- 市场是最自由的地方；
- 可以随意演练人生；

- 在任何时候都有操作机会；
- 爱上了就要负责，无论对错都毫无怨言；
- 爱上了就要负责到底（这里特别强调对亏损负责到底，这种负责是发自内心的负责）；
- 想要做一个负责人的人就要有很好的自律品质；
- 向赢家一样思考和学习；
- 要有安全措施，意外一定会发生；
- 适应市场规律，顺势而为；
- 在成功时要学会激流勇退，适可而止。

（3）进行趋势交易

趋势交易的周期一般都比较长，大多是对一个周期的预判，这个周期可以很长，比如10年或20年，也可以很短，比如1年或半年。每个周期都有很多判断方法，比如生育高峰期、建筑物使用寿命、执政党执政时间等。虽然判断周期的方法很多，但是在一个完整经济周期内的资本流通方式大体却是相同的，这种相同也为我们指明了大概的交易方向，如图4-7所示。

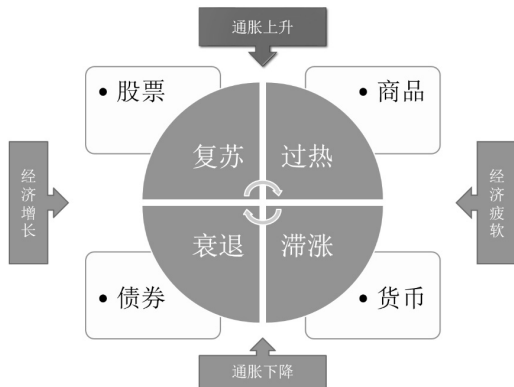


图 4-7 经济周期内的资本流通

除周期判断方法外，还可以通过技术指标来判断，判断的方法主要是看变化前后的比率。比如，二级市场中被广泛采用的是通道突破方法，就是当价格突破布林线或 CCI 时就开始买卖。

趋势交易的周期一般比较长，如果是期货，则在一个主力合约快要到期时，一般会以当前价位平仓，然后用同样的方向建仓顺延到下一个主力合约中。

（4）进行日内交易

日内交易的买卖非常灵活，几乎可以做到随买随卖。我国股票除港股外，目前还不可以日内交易。日内交易大部分挣得都是波动率的钱，许多日内交易都是不过夜的，一般会在当日进行结算，资金量一般也不大。进行日内交易的大部分都是投机者，投机者更看中短期的利润，虽然每笔的利润不是很多，但是当交易的频次足够多时，也就相当于自己拥有了较大的本金。日内交易对行情的实时性要求比较高，一般至少是分钟或逐笔级别的数据。进行日内交易的散户和游资比较多。

（5）高频交易

高频交易与其说是一个策略，还不如说是一项技术活。高频交易的好处在于速度快、频次高。速度快主要体现在行情、计算和排队上。光纤由于折射的原因肯定没有电磁波快，而局域网也一定快于城域网，城域网则快于广域网，更何况我国还有不同的运营商，到了港交所还有一部分路由，所以大部有实力的高频交易公司都高价租用交易所附近的机房。

高频交易中的快的好处就是，同样一笔交易，你能排到前面，就可以比别人先成交。就算是难以买卖的涨跌停股票也是有交易的，只不过交易量很少而已。

高频交易有一个致命的缺点，那就是佣金太高。由于交易的频次过高，比如千分之二的价差就进行买卖，真正挣到的钱简直微乎其微，大部分利润都给了交易所和证券公司，不过从事高频交易的基金公司都会与证券公司商谈，可以把佣金的价格压下来，当然最好拥有自己的交易席位，像西蒙斯那样。

5. 复盘

前面我们说的都是关于策略的部分，实际上复盘也是一个交易员需要

面对的工作，由于复盘不是我们策略中的主要工作，所以只是简单地提一下，下面以股票为例。

- (1) 检查账户情况并记录自己的操作，必要时写下心得，比如为什么买、为什么没买。
- (2) 查看主要指数的今日走势，重点看量价和走势，注意关键点位。
- (3) 检查所关注的股票是否有新的信息发布。
- (4) 检查自己的系统，包括自身状态，看明天是否能够继续交易。
- (5) 检查是否有需要与客户交流的内容。

4.1 基本面分析

基本面分析主要是从股票（商品）的基本信息来判断股价未来的走势，一般侧重于对经济和政治数据的透彻分析。股票的基本信息主要包括宏观经济情况、公司主营业务所处行业信息、公司业务与本行业的竞争水平和公司内部管理水平等，有时还把财务信息纳入公司的基本信息范围。

基本面分析可以很简单，如果特别相信某个评级公司的评价，那么就可以完全按照其评级进行操作。不过就目前看，港股普遍达不到评级公司的期望值。

基本面分析也可以很复杂，比如把政治周期模型、标准普尔指数、道琼斯指数、上证指数、恒生指数、消费者指数和我国 GDP 环比相结合，指数上涨就买入，指数下跌就卖出。

同时，基本面分析可以很离谱，比如当你发现身边的人都开始炒股时，你却决定退出股市；当大家都不“玩”股票的时候，你却重仓买入。甚至还有一个有趣的裙子理论，交易员没事的时候去大街上看美女们的裙子，如果发现裙子过短，则开始卖空股票；如果发现裙子过长，则开始买入股票。

基本面分析还可以很科学，股票根本的价值在于其分红，没有分红的股票算不上是好股票，有分红的股票一定要买入。

究竟哪种对呢？哪种更有道理呢？笔者个人觉得，每一种基本面分析方法都有它的可取之处，这种可取之处更多的在于交易员对数据的深层次理解。

就拿最荒唐的裙子理论来说，秋天穿的裙子必然比夏天穿的裙子长，难道我们就应该冬天买入股票、夏天卖出股票吗？裙子理论的本质一定不是这样的，裙子理论是说，当女人们衣着比较暴露时，说明经济已经过热了，经济过热了就要回归理性，所以这时逐渐卖出股票是正确的选择，正所谓饱暖思淫欲。基本面分析不能太过教条，我们不能因为窗外刮起了大风就买多农产品，因为你不在农产品主产区，你家窗外刮大风和农产品主产区没有必然联系。

由于股票是一直长期存在的金融产品，所以价格就会有起伏，这种价格起伏在较大的范围内来看就像波浪一样，只不过有时波浪大、有时波浪小而已。于是很多人就喜欢分析波浪的转折点，研究是什么原因造成了局势的反转。我国有句话叫作物极必反，似乎很能说明这个问题。我们知道，任何事物都有泡沫，关键是看这个泡沫的大小和人们对它的预期。比如荷兰的郁金香事件、英国的南海危机、美国的大萧条和 2008 年的金融危机。

刚才讲的裙子理论和经济周期其实都是为了方便预测这些大波浪的。从心理学的角度看，这些预测还是有一定道理的，如图 4-8 所示。

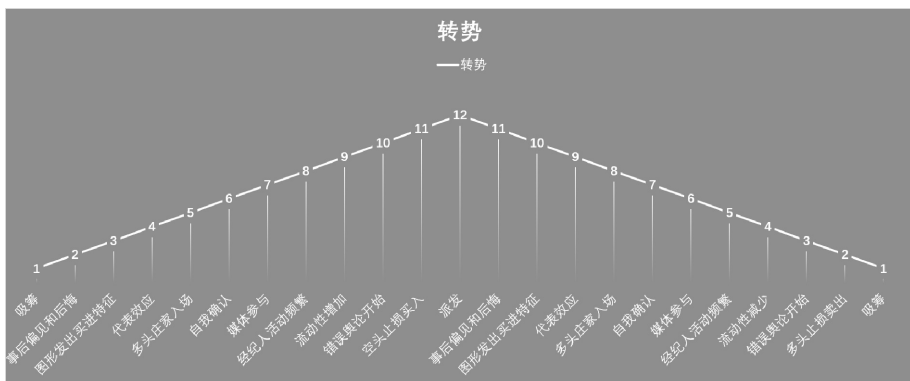


图 4-8 转势

从刚开始最简单的吸筹开始，到越来越明确的信号乃至庄家入场都是一个不断强化的过程，当媒体和所谓的经济学家也参与进来时，经济就开始过热了，如果空头无法熬过这一关，则只能止损离场（这时确实非常难熬，因为经济一旦过热就会成指数级增长），当空头止损、多头盈利时就是转势开始的时候，这时大空头们开始建仓，聪明的人也开始反手做空，于是新一轮波浪又开始了，只不过这次是以空头为准。

不知道读者注意到没有，转势的关键是有空头止损或多头止损才能开启新一轮循环。那么，我们怎么判断这个信号呢？

- (1) 加速和放量，比如新买家介入。
- (2) 趋势破坏，如双头顶。
- (3) 突然的变化，如急跌。
- (4) 缺乏广度，个股已经不跟大盘走了。
- (5) 特殊图形，不好理解的图形初现。

当然，如果你完全是一个地道的价值投资者，可以完全不关心是否转势，只要继续跟着企业分红就好了。价值投资者比较关注如图 4-9 所示的几点。



图 4-9 价值投资者的关注点

前面讲解了基本面分析，下面列举一个比较标准的基本面量化的例子。假设我们最近一直关注食品饮料行业（比如康师傅或达利食品）或白糖期货，那么就需要了解原材料现货市场的供求关系，比如淀粉和白糖。现在我们只分析白糖。

一般对于供给而言，商品供给的增加会引起价格的下跌，供给的减少会引起价格的上涨。对于需求而言，商品需求的增加将导致价格的上涨，而需求的减少将导致价格的下跌。也就是说，供大于求时，白糖价格下跌；供小于求时，白糖价格上涨。

那么我们再想一下，白糖价格上涨就相当于原材料价格上涨，如果原材料价格上涨，在商品价格不变（商品的价格在一定的时期内是固定不变的）的情况下就是收入下降，而收入下降时股票的评级就会下降。

也就是说，原材料价格上涨时，股价下跌；原材料价格下跌时，股价上涨。

（1）供给分析

既然刚才讲了原材料价格涨跌与供给有关，那么就先看看白糖的供给。世界食糖产量为 1.3 亿~1.50 亿吨，产量超过 1000 万吨的国家和地区有巴西、印度、欧盟和中国等，其中巴西的产量超过 2000 万吨。巴西、欧盟、泰国是世界食糖主要出口国家。巴西的糖业直接影响到国际食糖市场走向，其产量和供应量对我国白糖市场的影响力也比较大。

（2）需求分析

分析完了白糖的供给，我们还要看一下白糖的需求，世界食糖消费量约 1.35 亿吨，消费量较大的国家和地区有印度、欧盟、中国和巴西等。从近几年消费的情况来看，印度食糖年消费量维持在 2400 万吨左右，欧盟食糖年消费量维持在 1500 万吨左右，我国食糖年消费量增长至 1500 万吨左右，巴西食糖年消费量在 1000 万吨左右。我国白糖的需求量在逐年上升。

（3）进出口分析

接着进行白糖的进出口分析。我国白糖的进口量远远大于出口量，平

均每年进口食糖约 300 万吨，而出口量不到其十分之一，因此白糖进出口对我国市场的影响很大。白糖进口会增加国内供给数量，导致价格下降。由于进出口协议的关系，我国生产的大部分食糖还是自给自足。

我国白糖的出口量不多，每年都有库存，库存的作用主要是调剂余缺。一般来讲，库存水平提高时，供给宽松；库存水平降低时，供给紧张。库存的影响力虽然很大，但是这种影响力也取决于库存总量和国家介入的频次。库存总量越大、国家介入市场频次越高，价格起伏也就越平稳。

（4）气象分析

白糖是名副其实的农副产品，现在的农业绝大多数还是需要靠天吃饭，由于我国的白糖大部分都是蔗糖（只有很少的一部分甜菜糖生长在新疆和黑龙江地区，绝大部分甘蔗都生长在广西、广东一带），所以我们更加关注广西和广东地区的天气。甘蔗的生长期比较长，任何干旱、洪涝、大风、冰雹、低霜冻等天气都对其有灾害性的影响，而且这种影响一旦形成便是无法挽回的，只能不断地减产减收。比如，当广西崇左（我国甘蔗主产区）地区发生巨大台风时，那么我国当年的白糖产量就不会太高。

（5）季节分析

甘蔗是农作物，是农作物就有季节性特征。在我国，每年的 10 月至次年的 4 月为甘蔗集中压榨时间，由于白糖集中上市，造成短期内白糖供给十分充足；随着时间的推移和持续不断的消费，白糖库存量越来越少，而价格也往往随之发生变化，具有比较明显的季节性特征。

（6）替代品分析

分析白糖时还有一个重要的因素就是替代品，甜味剂就是白糖的主要替代品，它的使用减少了白糖的正常市场份额，对白糖的供给、价格都有一定的影响。当然，由于替代品所占的比重较小，所以对白糖实际的价格影响不大，但是如果你分析的企业非常依赖甜味剂，就不得不多分析一下甜味剂了，至少要知道甜味剂有哪些，比如淀粉糖、糖精和甜蜜素。如果市面上出现了一种新的制糖工艺，那么对应的食品企业的股

价就应该上涨（提高工艺水平，原材料价格下跌，食品价格不变，收益上涨，股价上涨）。

（7）其他分析

除前面讲的这些因素外，作为一名交易员，还要紧盯国家政策法律和相关科技，比如科技进步使国家放开食品企业糖精的使用量，那么甜味剂的比重势必会增加，原材料价格下跌，股价上涨。

综上所述，食品饮料企业股票卖出模型的构建思路如下。

（1）引入数据：广西甘蔗主产区天气预报、食品行业监督新闻。

（2）卖出条件：如果广西甘蔗主产区的天气预报说要出现特大台风，或者食品行业监督类新闻报道该食品企业违规等，那么就要立刻卖出该企业股票。

基本面分析还有一个比较细致的工作就是量化，例如特大台风究竟有多大、报道企业行为恶劣究竟有多恶劣等。

4.2 K线

K 线也叫蜡烛图，源于日本，主要包括开盘价、收盘价、最高价和最低价。如果收盘价高于开盘价，则收阳；反之则收阴。其中，开盘价和收盘价用较粗的矩形表示，最高价和最低价用线表示。

K 线能够比较充分地表现出某个时间段的价格趋势，如果多根 K 线相连，甚至可以表现出一定的多空关系，因此很多人通过 K 线的形态进行选股。K 线如图 4-10 所示。

在 K 线中，红色表示上涨，绿色表示下跌。

根据几百年来人们的经验，可以把股票的涨跌尤其是趋势的反转汇总成几个大类的图形信号，比如上涨信号就包括捉腰带、母子线（孕线）、回马枪和锤子线等。

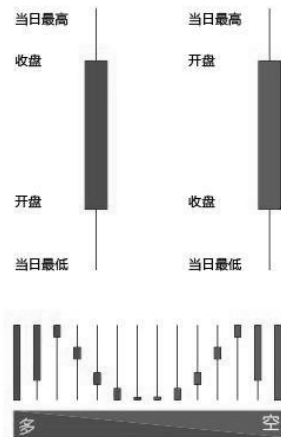


图 4-10 K 线（蜡烛图）

4.2.1 捉腰带

若一只股票的价格在下跌中，但在收盘时有一个比较大的阳线，这个阳线一般没有什么影线，这只股票的价格隔夜之后还能上涨一段时间，如果行情好的话，可能上涨得更多一些，有人将这种形态叫作捉腰带，即捉住腰带跟着上涨的意思。如果最近行情是上涨的，则效果更加明显。一般这类大阳线收盘的 K 线次日初现十字星的可能性比较大，如图 4-11 所示。



图 4-11 捉腰带 K 线

4.2.2 回马枪

这种形态前面是一个大阴线，后面是一个大阳线，并且阳线收盘价格回穿过阴线的二分之一处，这种股票一般都会回调。

尽量在超过昨日 K 线实体的二分之一处，并且最好是在收盘附近进行股票买入操作，操作太早容易形成十字星 K 线，效果可能不太理想。不过如果是跳空低开的 K 线，即便是十字星也是相对比较好的，如图 4-12 所示，观察一下，尤其是在临近收盘处。



图 4-12 回马枪 K 线

4.2.3 锤子线

锤子线有两种颜色，实体部分可以是阳线，也可以是阴线，也就是说无论是涨还是跌，都可以形成锤子线。锤子线最明显的特征就是有较长的下影线，实体较小并处于 K 线图的顶端位置，一般下影线的长度要大于实体长度的两倍以上。一般把位于行情底部的阳线收尾的 K 线叫作锤子线，锤子线说明行情已经接近地面了，不能再跌了，行情即将发生变化，应该会上涨（上涨居多）。反之，位于行情顶部的阴线收尾的 K 线则是上吊线，行情一般会下跌。我们这里只说锤子线，如图 4-13 所示。



图 4-13 （绿）锤子线 K 线

锤子线模型的构建思路如下。

引入数据：最近两周的历史数据和实时行情数据。

买入条件：

- (1) 连续 1 周收盘价或开盘价低于昨日的收盘价与开盘价中的最高价，判断为下跌趋势；
- (2) 开盘价减去收盘价的绝对值便是实体线；
- (3) 判断开盘价是否大于收盘价，如果大于收盘价，则是绿色锤子线，否则是红色锤子线，红色锤子线一般比绿色锤子线更有说服力；
- (4) 如果是绿色锤子线，则收盘价减去最低价便是影线；
- (5) 如果是红色锤子线，则开盘价减去最低价便是影线；
- (6) 如果影线除以实体线大于或等于 2，则是锤子线，买入。

4.3 多因子

4.3.1 随机策略

我们第一次买卖股票时，可能无从下手，因为不知道未来会怎样，所

以对我们来说买多或买空的成功概率都是一样的。怎么办？反正都一样，那就抛硬币吧！是正面就买多，是背面就买空。港股虽然不允许直接买空，但是可以操作牛熊证，如果不喜欢牛熊证和涡轮，就可以选择不买，其实不买也是变相的买空，也就是说等它下跌到一定程度后再买入。这样通过抛硬币来决定买或不买的好处在哪里呢？从本质上说，只会给证券公司送佣金，捞不到半点好处，但是美国《旧金山纪事报》却通过做类似的事情盈利了，是怎么做到的呢？

他们是这样用大猩猩做选股实验的：他们让大猩猩随机选择写有股票代码的纸板，一共选择了 5 只股票，然后用大猩猩挑选的股票组合与《华尔街日报》中 8 位知名分析师挑选的 5 只股票相比较，在持有一段时间后，大猩猩随机抽取的股票的票面价值竟超过职业分析师选择的股票的票面价值。表面上看这似乎不太可能，感觉大猩猩运气太好了，但是后来人们在相继的操作中发现一个关键细节，那就是定期卖掉所持有的股票中收益率最差的一只，再让大猩猩随机选择一只股票来买。

也许有人会说，这太不靠谱了，股票价格上涨最关键的因素是企业的收益率，收益率上涨，股票价格一定上涨。

也有人认为大盘指数才是最关键的因素，大盘指数上涨，股票价格一定上涨，毕竟大盘指数是由众多个股决定的。

还有人认为股票价格应该受到资本资产的影响，可以根据资本资产的关系进行定价，也就是所谓的资本资产定价模型。

这时候财务人员说，什么都不如财务指标好使，财务指标才是最忠诚的伙伴。

大家各不相让，每个人都据理力争，最终也没有结果。既然大猩猩随机选股的效果都那么好，那么就按照每个人的理解，各自加入一个或多个因素，然后也像大猩猩那样不断地轮换吧！

于是大家都同意这个办法，各自确定其关键因素去了。这里就用财务的一些因素来看一下，多因素（多因子）模型应该如果选股。其实单单就财务数据而言，影响股价变动的因素就有很多，如图 4-14 所示。

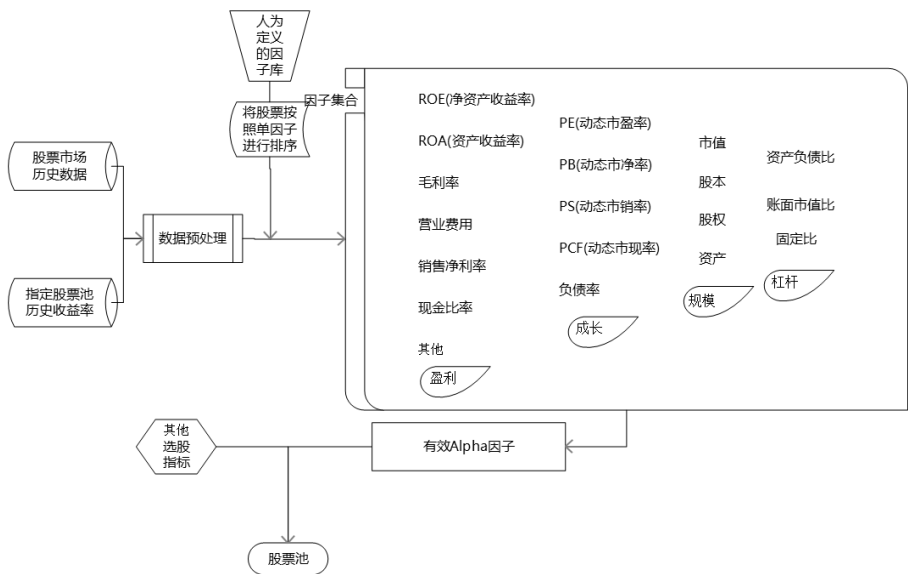


图 4-14 财务因素

4.3.2 多因子模型构建思路

引入数据：引入公司 10 年以上的历史财务数据、股票 10 年以上的历史数据和实时行情数据。

买入条件：

- (1) 根据财务因子进行不同的多因子组合，从单一因子到几十个因子，这种组合可能有成千上万个，不过没关系，我们就把每一个组合当成一个进程；
- (2) 随机选择 50 只股票进行买入；
- (3) 一个月后将表现最不好的 10 只股票剔除，被剔除的股票不再选择；
- (4) 不断地循环上述过程，直至历史数据全部遍历完；
- (5) 将总的收益与大盘指数进行比较，看看是否有超额收益；
- (6) 如果有超额收益，则可以进入模拟交易进行测试。

经验：关于多因子的组合，不仅是一个阶乘那么简单，还可以把不同的财务数据进行对比，得出新的因子。

根据经验，3~5 个多因子是比较合适的，过少容易过拟合，过多容易不收敛。

4.4 小海龟

当进行了一段时间的交易后，你会发现，怎么大部分时间股票都在一个区间内波动呢？一旦突破了某个点后，股价就像是脱缰的野马一样不受控制地按照趋势发展下去，于是人们就想，可以根据前一日收盘价设置一个价格范围，在这个价格范围内低买高卖，一旦突破一个上限后，就变成长期投资，平时就靠日内的波动率挣钱。这样可以画出五条线，中间的这一条线是前一日的收盘价，然后在前一日收盘价的基础上画两条区间线，接着在区间线的两边再画两条待突破线。这里有一个关键点就是买空线和趋势线之间是相互冲突的，所以一般趋势线的设置是以历史价格和概率来确定仓位比重和位置的，也就是说在这个策略组合中，如果历史数据表现证明趋势更明显，那么趋势线的仓位就要比日内波动率策略的仓位重，反之亦然。由于趋势线也是一条止损线，所以止损线的设置要注意假突破的问题，假突破是趋势策略最需要刻意避免的，一般情况下人们会根据最近一段时间的真实波动率来设置一个突破的区间，这五条线如图 4-15 所示。

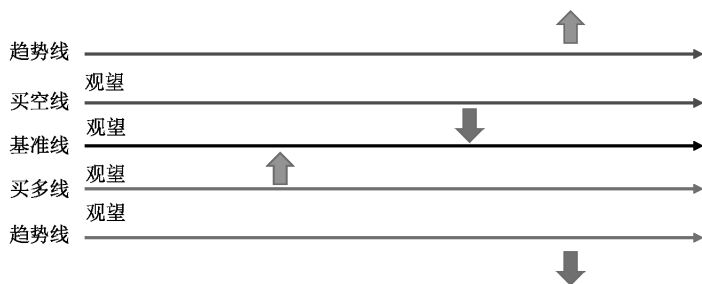


图 4-15 趋势反转图的五条线

另一个与之比较相似的策略是 RBreakePlusI 策略。它根据上一个交易日的收盘价、最高价、最低价数据和三个自选参数 f_1 、 f_2 、 f_3 ，通过一定的

方式计算出从大到小的6个价位，即突破买入价、观察卖出价、反转卖出价、反转买入价、观察买入价和突破卖出价，其中：

观察买入价 = 最低价 $-f_1 \times$ (最高价 - 收盘价)；

观察卖出价 = 最高价 $+f_1 \times$ (收盘价 - 最低价)；

反转买入价 = $(1+f_2)/2 \times$ (最高价+最低价) - $f_2 \times$ 最高价；

反转卖出价 = $(1+f_2)/2 \times$ (最高价+最低价) - $f_2 \times$ 最低价；

突破买入价 = 观察卖出价 $+f_3 \times$ (观察卖出价 - 观察买入价)；

突破卖出价 = 观察买入价 - $f_3 \times$ (观察卖出价 - 观察买入价)。

改变参数 f_1 、 f_2 、 f_3 的大小可以调节6个价格之间的距离。可以利用分钟线，根据收盘价在6个价格区间的上下波动来买卖。 f_1 、 f_2 、 f_3 经典的参数是： $f_1=0.35$ ， $f_2=0.07$ ， $f_3=0.25$ 。

关于日内策略还有一个经典的叫作空中花园策略，这个策略主要根据开盘的第一分钟来决定买卖，是一种很简单的入场方式，当然出错的概率也最高。也就是说，开盘第一根K线是收阳还是收阴，是判断日内趋势可能运动方向的标准，在当天开盘高开或低开时更有效。

空中花园策略在当天开盘高开或低开时使用，即当开盘价大于或等于昨天收盘价乘以1.01，或者开盘价小于或等于昨天收盘价乘以0.99时来决定买卖。其中：

上轨 = 第一根K线的最高价；

下轨 = 第一根K线的最低价；

当价格突破上轨时，买入开仓；

当价格跌穿下轨时，卖出开仓。

实际上，当天大幅高开(>1%)，可搏高开低走，反之亦然。

1. 小海龟策略

在经典策略中，海龟策略是非常有名的。海龟策略并不是一种仿生学，

只是以两个著名的交易员赌约所在地而命名，他们争论的地点在一家海龟养殖场，这就是海龟策略名称的由来。两位交易员争论的焦点是“优秀的交易员是否是天生的”，结果是天赋和努力都很重要。不过理查德·丹尼斯（他认为优秀的交易员不需要天赋）的策略却因为这件事得以广泛流传。

海龟策略从本质上讲是一个趋势跟随模型，通过唐安奇通道等突破方法确定入场和离场信号。海龟交易法则与其说是策略，不如说是一套系统，它不仅决定买卖什么、买卖多少、什么时候买卖、什么时候止损，还决定具体怎么买卖。

在海龟策略中，主要通过唐安奇通道的突破来进行买卖。唐安奇通道首次引入了上线、中线和下线的概念。

上线 = Max （前 N 个交易日的最高价）

下线 = Min （前 N 个交易日的最低价）

中线 = $(\text{上线} + \text{下线}) / 2$

每个交易日结束之后更新当天的数据。

这里 N 一般默认取 20，不过具体数值还要根据个人的经验来确定。

除唐安奇通道外，布林线通道的突破也很常用。布林线通道同样分为上线、中线和下线，如图 4-16 所示。



图 4-16 布林线通道

我们知道，通道突破一般都属于趋势交易，因为把通道用于短周期（比如日内）的效果非常不好，所以这里做了变通，日内交易仍然采用波动率的策略，我们把它形象地称为小海龟。有大量的小海龟，每一个小海龟都是一笔相对独立的资金。这种策略讲究分布式轻仓上阵，可以同时两个方向做交易，哪个方向好就先锁仓，也可以只做回调，比如在趋势上做对冲，假设有10%的仓位做多，那么小海龟最多可以10%做空，如果当天没有时间继续完成交易，感觉有利可图就可以结束当日交易，具体做法因人而异，总之就是积少成多，整体收益还是不错的。

锁仓时可以通过均线锁定，解锁后一定要立刻在合适的位置（均线、开盘价等位置）对冲一下，以免剧烈波动。

为了减少滑点，可以尽量在震荡时买入，要注意的是，这是大概率事件，一旦发现趋势错误，一定要及时锁住，然后使用通道突破的方法进行观察。就像小海龟，每天都要回家，尽量上午就到家，否则时间就不够用，小海龟可能回不到家，在外面游荡是很危险的，小海龟尽量不要在外面过夜。上午平仓一个股票之后，可以利用下午的时间反向平仓另一个股票。

当然对小海龟来讲，离家越远、天越黑往往食物就越多，但是危险也越大。为了能够健康地“活”下去，一定要谨慎。

小海龟和海龟的趋势交易是互补的，其主要思想很简单，就是顺着趋势去挣趋势中波动率的钱。

2. 小海龟模型的构建思路

引入数据：股票或期货历史数据，以及实时行情数据。

买入条件：

- （1）根据历史数据设置通道（布林通道或唐安奇通道）；
- （2）根据前一日收盘价设置日内交易的波动率区间；
- （3）在波动率区间内低买高卖；
- （4）当波动率形成通道突破时，进行大仓位的趋势交易；

(5) 波动率策略，日内平仓或锁仓。

4.5 高频交易

1. 高频交易策略

高频交易在我并不流行，主要是我相关法律的限制，相关法律认为高频交易有损交易的本质，甚至造成不必要的动荡，比如，一旦高频交易公司的计算机发生故障，成千上万错误的买盘指令将充斥整个市场，造成价格的巨大波动，形成闪电崩盘，甚至使得某一个交易所的系统崩溃。

不过，对于交易者和证券公司来说，高频交易却有着不小的魔力。那么，究竟什么是高频交易呢？高频交易是指那种能够提高资金周转率的交易。这种技术一般都需要借助强大的计算机系统和复杂的运算才能完成。提高资金周转率的方法除增加杠杆外，就只能靠增加交易频次了。高频交易主要是通过极短的市场变化中进行买卖，从而获利。为了追求这种高频次的交易，普通技术已经不实用了，很多这种类型的基金公司都聘请了计算机、物理和数学方面的专家。所以，如果你发现一家基金公司有很多计算机高手和数学天才，那么他们八九不离十就是高频交易公司了，这方面做得最好的就是西蒙斯的文艺复兴科技公司。

抛开政策层面的因素，下面简单介绍一些常用的高频交易策略，这些策略主要包括统计套利、时间套利和低延迟策略等。

统计套利非常常见，就是通过数学统计的方法发现那些成功概率比较高的机会，比如我们发现两只食品股票的走势常年基本保持一致，于是就可以设置一个偏离率来进行买卖，大于某个值就买，回到正常范围内就卖出，通过计算机技术，系统可以同时统计数千只股票的偏离率，从而获利。

时间套利需要一定的爬虫技术和自然语言处理技术，同样一则消息人类阅读是比较慢的，而计算机阅读就可以在毫秒之间进行判断，谁第一时间掌握了消息，谁就更有机会获得收益。

低延迟策略是一个技术活，这项工作需要物理专业和通信工程的知识，国外的一些基金公司甚至利用微波来代替传统的独立光纤，因为微波在空气中的传播速度，相较于真空中的光速只慢了1%，而光在光纤中的传播速度却要减慢30%左右。

2. 高频模型的构建思路

引入数据：企业官方网站数据和关键词感情色彩（利空或利多）数据库。

买入条件：

- （1）实时爬取企业官方网站的数据；
- （2）对照关键词词库，当企业官方网站的数据出现利好消息时买入。

3. 关于A股T+0

日内T+0是针对我国股票而言的，由于我国股票是T+1的，所以无法完成日内交易，为了解决这个问题，很多有经验的人都开始变向地做起了T+0（以下简称“做T”）。做T的基本思路非常简单，就是高抛低吸。做T是指之前就已经持有了某只股票，比如现在有100股万科A股票，然后又买了100股万科A股票，那么就可以卖出之前买的那100股万科A股票。

做T的目的是降低股票持仓成本，用少量的资金追求每日波动的利润。做T可以挣趋势的钱，也可以挣波动率的钱，或者两者都参与，唯一的要求就是永远持有一定比例的股票。

还有一些操作方法是把一笔资金分成10份或20份，资金的份数和某只股票历史的波动频率和范围有关。比如有些人把一日的波动率按照 $\pm 2\%$ 分为10个价位，然后根据金字塔建仓的方法，每个价位都有一笔买入或卖出的操作，每天结束都保持一定的仓位持续持有这只股票，只不过这些股票不是在同一天、同一个价位买入的，很可能是10种不同的价格，所以在次日交易的时候，看上去高频一些。

这种操作并不鼓励，因为我国市场并不是特别欢迎类似的操作手法，不过作为策略最好还是了解一下。

5

第 5 章

项目拓展

5.1 模拟交易系统

我们评价一个策略的好坏，一般都是通过模拟交易来衡量的。模拟交易的好处是既没有未来函数，也不用担心亏钱。现在很多软件都提供这个功能，比如某个交易软件，用户注册一个账户，然后就可以进行模拟买卖了，不过这些模拟交易一般都是需要手动操作的，而系统为了满足大量策略进行模拟测试，就需要一套自动化程度较高的模拟交易系统，这个系统不用太复杂，能够根据行情模拟买卖就可以了。模拟交易系统的开发流程如图 5-1 所示。

1. 选择交易接口

选择交易接口，这里主要是指选择股票或期货行情数据接口。提供行情数据接口的网站和软件很多，收费的数据质量自然好，这里所说的数据质量主要是指实时性和逐笔交易；免费的数据一般都是分钟级别的数据，这里的分钟数据一般都是数据服务商自己合成的，所以在数据精度上可能不如 Tick 级别的精细，但是对于模拟交易来说已经足够了。

本书采用新浪网站提供的免费行情数据接口，参考网址是 <https://hq.sinajs.cn/>。

获得股票的请求示例为 <https://hq.sinajs.cn/list=sz000002>。

获得期货的请求示例为 <https://hq.sinajs.cn/list=RU0>。

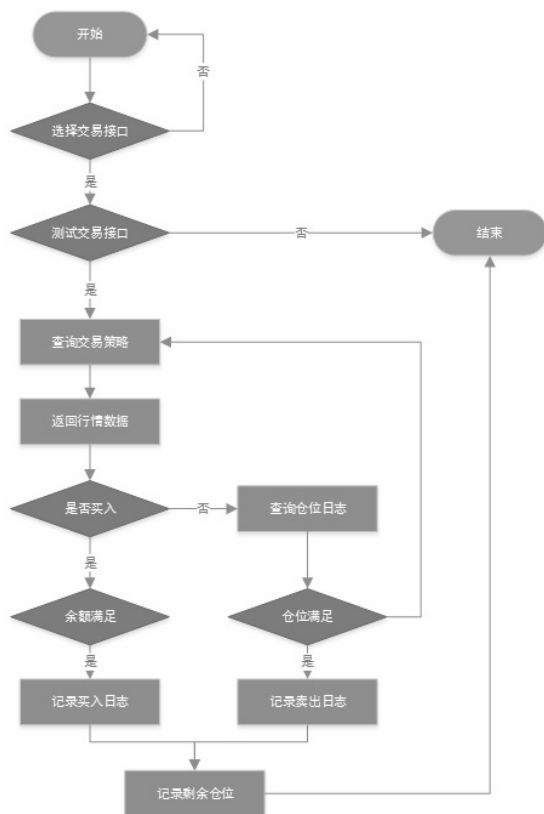


图 5-1 模拟交易系统的开发流程

2. 测试交易接口

由于新浪网站提供的接口使用了 HTTPS 协议,所以我们使用 PHP 的 curl 方法进行请求,尤其要注意 curl_setopt 函数中 CURLOPT_SSL_VERIFYHOST 和 CURLOPT_SSL_VERIFYPEER 设置为 false 时才能起作用。另外,不要忘记开启 PHP 的 openssl 扩展。新浪网站行情数据接口的测试代码如下。

```
sina_test.php
<?php
//新浪网站股票行情数据接口
$gp ='https://hq.sinajs.cn/list=sz000002';
```

```
echo getURL($gp);
//新浪网站期货行情数据接口
$qh = 'https://hq.sinajs.cn/list=RU0';
echo getURL($qh);
//用 curl 方法请求 HTTPS
function getURL($url){
    $curl = curl_init($url);
    curl_setopt($curl, CURLOPT_FAILONERROR, true);
    curl_setopt($curl, CURLOPT_FOLLOWLOCATION, true);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, false);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    $result = curl_exec($curl);
    curl_close ($curl);
    //转码为 UTF-8
    $json = iconv('GB2312', 'UTF-8', $result);
    return $json;
}
?>
```

3. 交易指令

这里可以先写一个简单的下单命令，至于为什么会形成这样的交易指令是由交易员和策略模块决定的，模拟交易系统只需要按照这些命令进行准确无误的交易就可以了，实际上交易员也不会把他的交易策略告诉你，因为他还不想与别人分享他赚钱的秘密。

因此，我们的工作也变得简单了，只需在 `ai_game` 数据库中创建一个股票交易指令表，其表结构如表 5-1 所示。

表 5-1 股票交易指令表 `stock_dealer`

字段	类型	长度	默认	注释	主键
id	Int	10	0	指令 id	*自增长，主键
type	tinyint	1	0	交易方向，0 为卖，1 为买	
code	char	20	Null	交易代码	
starttime	int	10	0	开始时间	
endtime	int	10	0	结束时间	
number	int	10	0	交易手数	

续表

字段	类型	长度	默认	注释	主键
price	float(3 位小数)	10	0	交易价格	
addtime	int	10	0	交易时间	
disabled	tinyint	1	0	是否完成, 0 为未完成, 1 为已经完成, 2 为过期作废	

交易指令表中有开始时间和结束时间,但并不是说交易指令表中的指定时间内就可以进行交易,还要看交易所是否开盘,以及这只股票是否停牌。交易所是否开盘这里可以调用上证指数来获得,接口为 https://hq.sinajs.cn/list=s_sh000001,而具体某只股票是否停牌可以通过其返回值来确定,如果返回值为 0,就表示停牌,比如乐视网 https://hq.sinajs.cn/list=s_sz300104 的返回值为:

```
var hq_str_s_sz300104="乐视网,0.00,0.000,0.00,0,0";
```

与我们早先测试的接口不同,这次使用了简约行情数据接口,实际上就是在原来接口中股票代码的前面加一个前缀“s_”,sh000001 是完整的上证指数,而 s_sh000001 则是简约的上证指数。简约数据的返回格式为:

```
var 代码="证券简称,最新价,涨跌额,涨跌幅,成交量,成交额"
```

下面以万科 A 股票为例,返回其最新价、涨跌额、涨跌幅、成交量和成交额,测试代码如下。

```
sina_sz000002.php
<?php
//万科 A 股票数据接口
$gp = 'https://hq.sinajs.cn/list=s_sz000002';
$s_wankea = getURL($gp);
$gp = 'https://hq.sinajs.cn/list=sz000002';
$swankea = getURL($gp);
function getURL($url){
    $curl = curl_init($url);
    curl_setopt($curl, CURLOPT_FAILONERROR, true);
    curl_setopt($curl, CURLOPT_FOLLOWLOCATION, true);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, false);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
```

```

        $result = curl_exec($curl);
        curl_close ($curl);
        //转码为 UTF-8
        $json  = iconv('GB2312', 'UTF-8', $result);
        //去掉换行符
        return  str_replace(array("\n","\r"), "", $json);
    }
    ?>
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>万科 A</title>
</head>

<body>
<script type="text/javascript">
eval('<?php echo $s_wanke; ?>');
//分成数组
sz000002 = hq_str_s_sz000002.split(",");
txt = sz000002[0];
txt += " 最新价:" + sz000002[1];
txt += " 涨跌额:" + sz000002[2];
txt += " 涨跌幅:" + sz000002[3];
txt += " 成交量:" + sz000002[4];
txt += " 成交额:" + sz000002[5];
alert(txt);
.....

```

在上面的代码中，getURL 函数的末尾多了一个删除换行符的功能，这个换行符虽然方便直接生成段落文本，但是需要使用 eval 函数运行 JavaScript 脚本，所以先删除了。

一般来说，这样简单的下单简约数据接口就够了，不过由于很多策略比较复杂，所以这里也顺带介绍一下其完整的数据格式。完整的数据格式主要是为了方便绘制 K 线和显示五档买卖行情的，再加上日期和时间等一共有 32 个字段，下面展示一下其完整的数据结构，代码如下。

```
sina_sz000002.php
.....
$gp = 'https://hq.sinajs.cn/list=sz000002';
$wankea = getURL($gp);
.....
//完整数据
eval('<?php echo $wankea; ?>');
//分成数组
sz000002 = hq_str_sz000002.split(",");
txt = sz000002[0];
txt += " 今日开盘价:" + sz000002[1];
txt += " 昨日收盘价:" + sz000002[2];
txt += " 当前价格:" + sz000002[3];
txt += " 今日最高价:" + sz000002[4];
txt += " 今日最低价:" + sz000002[5];
txt += " 竞买价:" + sz000002[6];
txt += " 竞卖价:" + sz000002[7];
txt += " 成交手数:" + sz000002[8];
txt += " 成交金额:" + sz000002[9];
txt += " 买一手数:" + sz000002[10];
txt += " 买一报价:" + sz000002[11];
txt += " 买二手数:" + sz000002[12];
txt += " 买二报价:" + sz000002[13];
txt += " 买三手数:" + sz000002[14];
txt += " 买三报价:" + sz000002[15];
txt += " 买四手数:" + sz000002[16];
txt += " 买四报价:" + sz000002[17];
txt += " 买五手数:" + sz000002[18];
txt += " 买五报价:" + sz000002[19];
txt += " 卖一手数:" + sz000002[20];
txt += " 卖一报价:" + sz000002[21];
txt += " 卖二手数:" + sz000002[22];
txt += " 卖二报价:" + sz000002[23];
txt += " 卖三手数:" + sz000002[24];
txt += " 卖三报价:" + sz000002[25];
txt += " 卖四手数:" + sz000002[26];
txt += " 卖四报价:" + sz000002[27];
txt += " 卖五手数:" + sz000002[28];
```

```
txt += " 卖五报价:" + sz000002[29];
txt += " 日期:" + sz000002[30];
txt += " 时间:" + sz000002[31];
alert(txt);
</script>
</body>
</html>
```

如果不想使用这种 JavaScript 格式,还有一种简捷的方法,接口还提供了一种文本格式,想要返回文本格式只要在接口中添加 `format=text` 参数就可以了。接口格式的代码如下。

https://hq.sinajs.cn/format=text&list=s_sz000002

`s_sz000002=万科A,23.27,0.05,0.22,254769,59047`

上面的文本只要使用逗号分隔符就可以简单地转换成数组了,比如 PHP 的 `explode` 函数和 JavaScript 的 `split` 函数。

如果还想研究这些免费的行情接口,则在缺少相关文档的情况下,最好通过加载了 <https://hq.sinajs.cn/> 的页面进行分析,毕竟浏览器的“开发者工具”是一个很不错的工具。

这里推荐两个参考页面方便读者深入研究,股票参考网址是 <http://finance.sina.com.cn/stock/>,期货参考网址是 <http://finance.sina.com.cn/futuremarket/>,或者看一下 `sina_ru0.php` 文件中的源码,其基本原理与股票类似,感兴趣的读者可以参考。本系统默认以股票交易为主,所以就不过多介绍期货交易了。

查询交易指令要遍历交易指令的起止时间是否在交易所开市的时间范围内,如果不在开始的时间范围内,即使模拟交易成功了也没有任何意义。交易指令遍历可以是一个总进程,也可以单独启动,为了说明方便这里采用单独启动方式。

对于有经验的交易员来说,在买卖股票时一般都是排队进行的,由于排队的人较多,所以不太可能马上就成交,如果想顺利买入,最好的方法是以卖家的价格买入、以买家的价格卖出,这种方法也叫滑点。滑点是交易佣金之外的另一项成本,不过一般小规模的资金也可以通过排队来减少

这部分成本。

另外还有一点需要说明，由于一次买入的数量可能比较大（或者成交量比较小），大于当前排队成交的数量，所以这时要向下一档进行买入，不过这里只看卖一档的，一般资金规模较小时，卖一档的数量就能满足测试需求了。

在实际操作中，由于资金量较大，所以可以把卖一档到卖五档的数量都计算进去，当然最好通过购买十档行情的数据接口一次性计算到卖十档。下面就来看一下具体实现过程。

首先是一个基本的设置，比如超时、数据库连接和表头，代码如下。

```
stock_sina.php
<?php
//导入配置文件
include 'config.php';
$type = array(0=>'卖出',1=>'买入');
set_time_limit(60000);
ignore_user_abort(TRUE);
$txt = '<table width="100%" border="1" align="center"
cellspacing="0" cellpadding="0">';
$txt .= '<caption>交易记录</caption>';
$txt .= '<tr><th>代码</th><th>交易方向</th><th>价格</th><th>数量
</th><th>时间</th><th>备注</th></tr>';
// 输出表头
echo $txt;
ob_flush();
flush();
```

然后遍历股票交易指令表，分成上午和下午，进行区间段的查询。A股早盘是 9:30 开盘、11:30 午休，午盘是 13:00 开盘、15:00 收盘。深圳股票交易所最后几分钟的收盘一般是无法交易的。我们通过修改 work.txt 文件的内容可以达到停止模拟交易的目的，比如，如果 work.txt 内容大于 1，则停止交易，代码如下。

```
stock_sina.php
.....
while(true) {
    // 是否停止
```

```

$work = file_get_contents('work.txt');
if($work > 0){
    goto stop;
}
//根据时间段查找交易指令
$date = date('Y-m-d');
$time = time();
$am_open = strtotime($date . ' 9:30');
$am_close= strtotime($date . ' 11:30');
$pm_open = strtotime($date . ' 13:00');
$pm_close= strtotime($date . ' 15:00');

if($time < $am_close && $time > $am_open){
    //早盘
    $sql = "SELECT * FROM `stock_dealer` WHERE `starttime`
< $am_open AND `endtime` > $am_close";
    $data = DataList($sql,$MYSQL);
}else if($time < $pm_close && $time > $pm_open){
    //午盘
    $sql = "SELECT * FROM `stock_dealer` WHERE `starttime`
< $pm_open AND `endtime` > $pm_close";
    $data = DataList($sql,$MYSQL);
}else{
    //闭市
    $data = 0;
}
if(!count($data)){
    goto stop;
}
    
```

接着判断是否能够满足交易，这里使用的是标准文本接口，即“<https://hq.sinajs.cn/format=text&list=>”。如果交易成功，则进行记录。说到记录，这里还有一个复杂的交易情况，比如部分成交。一般部分成交后，我们通常的做法是修改当前状态，剩余的未成交部分作为一个新的交易指令追加到记录中，同时还要修改原来交易指令中的交易数量。因为在实际交易中还有额外的佣金产生，所以我们默认是全额交易。

如果交易不成功，则继续轮询，遍历后，休眠 1 秒再次查询交易指令

表，代码如下。

```

.....
//循环判断是否交易成功
$i = 0;
foreach($data as $v){
    //这里是标准接口，默认全部成交
    $url = 'https://hq.sinajs.cn/format=text&list=' .
    $v['code'];
    $gp = getURL($url);
    $ar = explode(',',$gp);
    //判断交易方向
    $ok = 0;
    if($v['type'] == 0){
        //卖出
        if($ar[3] >= $v['price'] && $ar[8] >= $v['number'])
        { //以当前价格卖出
            $ok = 1;
        }else if($ar[11] >= $v['price'] && $ar[10] >=
        $v['number']) { //以买家价格成交
            $ok = 1;
        }else if($ar[1] == 0) { //停牌
            $ok = 0;
        }else{
            //滑点方法
        }
        //存入余额
        if($ok){
            //这里减去佣金
            $number = $v['number'] * $v['price'] - 10;
            $MYSQL->query("UPDATE money SET number =
            number + $number WHERE userid = 1");
        }
        }else if($v['type']==1){
            //买入
            if($ar[3] <= $v['price'] && $ar[8] >=
            $v['number']) { //以当前价格卖出
                $ok = 1;
            }else if($ar[21] <= $v['price'] && $ar[20] >=

```

```

$v['number']]{ //以卖家价格成交
    $ok = 1;
}else if($ar[1] == 0){ //停牌
    $ok = 0;
}else{
    //滑点方法
}
//存入余额
if($ok){
    //这里加上佣金
    $number = $v['number'] * $v['price'] + 10;
    $MYSQL->query("UPDATE money SET number =
number - $number WHERE userid = 1");
}
}else{}
if($ok){
    $id = $v['id'];
    $MYSQL->query("UPDATE `stock_dealer` SET `disabled`
= '1' WHERE `id` = $id");
    echo '<tr><td>' . $v['code'] . '</td><td>' .
$type[$v['type']]. '</td><td>' . $v['price'] . '</td><td>' .
$v['number'] . '</td><td>' . date('Y-m-d H:i:s') . '</td>';
    //这里是静态文件地址
    echo '<td></td></tr>';
    $i++;
}else{
    echo '<tr><td colspan="6">一共交易' . $i . '笔
</td></tr>';
}
ob_flush();
flush();
}
//休眠1秒
sleep(1);
}

```

最后输出结束提示符，顺便对过期的交易进行作废处理，代码如下。

```

stop:
//这里顺便进行处理。如果过了最后的时间还没有交易，就进行过期处理

```

```
$time = time();  
$MYSQL->query("UPDATE `stock_dealer` SET `disabled` = '2' WHERE  
`endtime` < $time AND `disabled` = '0'");  
//输出表尾  
echo '<tr><td colspan="6">已经停止交易</td></tr></table>';  
?>
```

关闭进程的代码更加简捷，其实就是通过传参来修改 work.txt 文件的内容，代码如下。

```
http://localhost/futu/stop.php?work=1  
stop.php  
<?php  
$id = @$_GET['work']?$_GET['work']:'0';  
file_put_contents('work.txt',$id);  
?>
```

4. 添加交易指令

在前面的代码中，交易成功时有一个余额的加减过程，其实主要修改的就是账户的余额，账户余额表（money）如表 5-2 所示。

表 5-2 账户余额表

字段	类型	长度	默认	注释	主键
userid	int	10	0	用户 ID	*
number	float(3 位小数)	10	0	账户余额	*

添加交易指令主要是为了验证余额是否充足，在仓位满足后增加一条记录，增加记录后要及时修改账户余额，代码本身并不复杂。以股票买入为例，如果以 POST 方式提交的表单包含 code，我们就认为有内容，否则不做记录。如果 type 字段的内容为 1，表示要进行买入操作，买入的时候要先计算余额是否大于或等于订单交易额，只有余额大于或等于订单交易额时，才能够下达交易指令，这时增加一条买入的交易指令，增加后要从余额中减去对应的交易额；反之，若 type 字段为 0，就进行卖出操作，卖出时需要计算已经成功买入的订单，并减去成功卖出的订单和挂单，如果剩余仓位大于或等于卖出的手数，就进行卖出记录，代码如下。

```
add_stock.php
```

```

<?php
//导入配置文件
include 'config.php';
//账户余额
$sql = "SELECT * FROM `money` WHERE `userid` = 1 LIMIT 1";
$data = DataList($sql,$MYSQL);
$point= $data[0]['number'];
if($_POST['code']){
    $code = $_POST['code'];
    $starttime = strtotime($_POST['starttime']);
    $endtime = strtotime($_POST['endtime']);
    $type = $_POST['type'];
    $price = $_POST['price'];
    $number = $_POST['number'];
    $txt = '';
    if($type){
        //买入，计算交易额是否满足要求
        $sum = $price * $number;
        if($point >= $sum){
            //可以交易
            $sql = "INSERT INTO `stock_dealer`(`type`,`code`,`starttime`,`endtime`,`number`,`price`,`disabled`) VALUES('$type','$code','$starttime','$endtime','$number','$price','0')";
            if($MYSQL->query($sql)){
                //减去余额
                $MYSQL->query("UPDATE money SET number = number - $sum WHERE userid = 1");
                $point = $point - $sum;
                $txt = '成功下买入单! ';
            }else{
                $txt = '买入下单失败! ';
            }
        }else{
            $txt = '账户余额不足! ';
        }
    }else{
        //卖出，返回该只股票的所有买卖记录和挂单情况
        $sql = "SELECT sum(`number`) AS `n` FROM `futures_dealer`

```

```

WHERE `type` = 1 AND `code` = '$code' AND `disabled` = 1";
    $buy = DataList($sql,$MYSQL);
    $sql = "SELECT sum(`number`) AS `n` FROM `futures_dealer`
WHERE `type` = 0 AND `code` = '$code' AND `disabled` = 1";
    $sell = DataList($sql,$MYSQL);
    $sql = "SELECT sum(`number`) AS `n` FROM `futures_dealer`
WHERE `type` = 0 AND `code` = '$code' AND `disabled` = 0";
    $look = DataList($sql,$MYSQL);
    //剩余仓位
    $space = $buy['n']-$sell['n']-$look['n'];
    if($space >= $number){
        $sql = "INSERT INTO `stock_dealer`(`type`,`code`,
`starttime`,`endtime`,`number`,`price`,`disabled`) VALUES('$type',
'$code','$starttime','$endtime','$number','$price','0');"
        if($MYSQL->query($sql)){
            $txt = '成功下卖出单! ';
        }else{
            $txt = '卖出下单失败! ';
        }
    }else{
        $txt = '仓位不足! ';
    }
    }
    echo $txt;
}
.....

```

为了简捷，我们把服务端和前端代码写到了同一个页面内。签单是一个专门用于下单的页面，包括股票代码、开始时间、结束时间、买卖方向、价格和数量等字段。如果需要，还可以提示用户当前的剩余额度，代码如下。

```

add_stock.php
.....
$starttime = date('Y-m-d');
$endtime = date('Y-m-d',strtotime('+1 day'));
?>
<!doctype html>
<html>
<head>

```

```

<meta charset="utf-8">
<title>股票添加页面</title>
</head>
<body>
<form id="f" action="add_stock.php" method="post">
<table align="center">
<caption>股票下单页面，账户余额：<?php echo $point;?></caption>
<tr><td>股票代码</td><td><input name="code" type="text" />
</td></tr>
<tr><td>开始时间</td><td><input name="starttime" type="text"
value="<?php echo $starttime; ?>" /></td></tr>
<tr><td>结束时间</td><td><input name="endtime" type="text"
value="<?php echo $endtime; ?>" /></td></tr><br>
<tr><td>买卖方向</td><td><label><input name="type" type="radio"
value="0" checked />卖出</label><label><input name="type"
type="radio" value="1" />买入</label></td></tr>
<tr><td>订单价格</td><td><input name="price" type="text"
/></td></tr>
<tr><td>购买数量</td><td><input name="number" type="text"
/></td></tr>
<tr><td colspan="2"><input type="button" value="提交"
onClick="isSubmit()" /></td></tr>
</table>
</form>
<script type="text/javascript">
//提交前的验证，比如资金和仓位是否满足下单要求
function isSubmit(){
    //'可以交给服务端验证，也可以用 Ajax 验证
    document.getElementById("f").submit();
}
</script>
</body>
</html>

```

5. 富途牛牛插件

前面讲解了基于 HTTPS 协议的免费行情接口的模拟交易，在实际项目中可能是收费版的，这时就需要账户名、密码和 Token 等字段了，不过具

体开发方法还是大同小异的。

开发时最好用券商提供的或交易所直接提供的 HTTP 接口，但是相对来说这些成本也是不低的，这里我们做一个稍微复杂一点的本地模拟交易接口。

由于法律的规定，国内并不鼓励自动化交易，所以在实际项目中，模拟交易的大多是港股。对于港股，国内一般都有 15 分钟的延迟，所以我们使用了一款专门交易港股的软件，这个软件叫作富途牛牛。虽然当初的目的是奔着高频交易去的，但是很多软件都不提供这种频次的接口。想使用富途牛牛软件交易港股，就需要用到港币，小规模的资金出境还是比较容易的。

首先要在富途牛牛上开个案，然后才能进行交易，交易一段时间后就有机会看到港股的十档行情。这里有必要对版本做一下说明，富途牛牛软件只有某些版本提供交易接口，所以要下载对应的版本。富途牛牛 3.3 版本之后，开始支持插件开发，接口规范详见 <https://github.com/szmile2008/FTPlugin.git>，但考虑到实现插件需要较深厚的 Windows VC 编程基础，用户使用多有不便，故官方提供了 socket 插件中间层，方便用户使用第三方工具如 Python 和 PHP 以脚本（如果只是模拟交易，也可以不使用脚本）的方式调用行情交易接口。

插件安装说明如下。

（1）如果当前的 Windows 富途牛牛的版本号低于 3.2，则双击 `ftnninstall.exe` 文件进行安装。

（2）`pluginserver.dll` 是已经编译成功的发行版本，如有需要，可自行编译。

（3）将 `config.ini` 和 `pluginserver.dll` 复制到 `ftnn` 安装目录下的 `plugin` 子目录中。

关于安装包，可以加“PC 牛牛抢先体验 QQ 群 438919070”或“FUTU 开放 APIQQ 群 108534288”下载对应的版本，安装完就不要更新了。

富途牛牛插件版打开后的界面如图 5-2 所示。

富途牛牛的开发文档还是比较完备的，由于这里是模拟交易，所以可以不通过 PHP 脚本而直接通过 PHP socket 编程。下面还是先测试一下吧，地址默认为 127.0.0.1，端口默认为 11111，发送一个简单的报文，测试代码如下。



图 5-2 富途牛牛插件版打开后的界面

```
futu_test.php
<?php
$ip = '127.0.0.1';
$port = 11111;
//创建 IPv4、TCP
$socket = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);
if( $socket == FALSE ) {
    echo '创建失败: ' . socket_strerror(socket_last_error());
} else {
    echo '开始';
}
// 连接
echo '连接 ' . $ip . ':' . $port . '<br>';
$result = socket_connect($socket, $ip, $port);
if ( $result == FALSE ) {
    echo '错误';
}

$in = '{"Protocol": "1005", "ReqParam": {"StockSubType": "1",
"Market": "1", "Stock Code": "00700"}, "Version": "1"}' . "\r\n";
if(!socket_write($socket, $in, strlen($in)) ) {
    echo '发送失败: ' . socket_strerror(socket_last_error());
}
```



```

} else {
    ob_flush();flush();
    echo "成功发送!<br>";
    echo '内容: ' . $in . '<br>';
}
$que = socket_read($socket, 8095,PHP_NORMAL_READ);
echo '返回: '.$que;
socket_close($socket);
?>

```

如果返回内容为如下字样就表示接口成功通过了，代码如下。

```

开始连接 127.0.0.1:11111
成功发送!内容: {"Protocol":"1005","ReqParam":{"StockSubType":
"1","Market": "1","StockCode":"00700"},"Version":"1"}
返回: {"ErrCode":"0","ErrDesc":"","Protocol":"1005","RetData":
{"Market":"1"," StockCode":"00700","StockSubType":"1"},"Version":
"1"}

```

下面介绍与报文相关的内容，这些在开发文档 OpenInterface/Json /PLS API 的说明文档中都能够找到，如表 5-3 所示。

表 5-3 协议号及其说明

	协议号	说明	协议号	说明
行情相关	1001	报价	1002	摆盘
	1005	订阅	1006	反订阅
	1007	查询订阅	1008	推送协议
	1010	分时	1011	最近 K 线
	1012	逐笔	1013	交易日
	1014	股票信息	1015	市场快照
	1023	批量报价	1024	历史 K 线
	1025	复权因子		
交易相关	6003	港股下单	7003	美股下单
	6004	港股订单状态更改	7004	美股订单状态修改
	6005	港股改单	7005	美股改单
	6006	解锁交易	7007	美股查询账户信息
	6007	港股查询账户信息	7008	美股查询订单列表
	6008	港股查询订单列表	7009	美股查询持仓
	6009	港股查询持仓	7010	美股查询成交记录
	6010	港股查询成交记录		

前面用到的 1005 号协议就是订阅的意思，富图牛牛的行情接口必须先订阅才能够使用和查看行情，而且有数量、频次等限制，特殊需求需要和商务人员进行沟通。

错误码及其说明如表 5-4 所示。

表 5-4 错误码及其说明

错误码	错误说明
0	无错误
400	未知错误
401	版本不支持
402	股票未找到
403	协议号不支持
404	参数错误
405	频率错误（请求过快）
406	订阅达到上限
407	未订阅
408	未满足反订阅时间限制
501	服务器忙
502	超时

我们这里只用到了行情数据，所以主要使用第 1001 号协议，即获取基础报价，代码如下。

```
$in = '{"Protocol":"1001","ReqParam":{"Market":"1","StockCode":  
"00700"},"Version":"1"}'\r\n";
```

上述代码返回值如下。

```
{"ErrCode":"0","ErrDesc":"","Protocol":"1001","RetData":{"Clo  
se":"135400","Cur":"135400","High":"137600","LastClose":"139800",  
"LotSize":"0","Low":"133800","Market":"1","Open":"136900","StockC  
ode":"00700","Time":"57600","Turnover":"3588168326322","Vol":"264  
41962"},"Version":"1"}
```

上述代码中的字段及其说明如表 5-5 所示。

表 5-5 字段及其说明

字段名	字段值	备注
Close	Int32	收盘价
High	Int32	最高价

续表

字段名	字段值	备注
LastClose	Int32	昨日收盘价
Low	Int32	最低价
Open	Int32	开盘价
Turnover	Int64	成交额
Vol	Int64	成交量
LotSize	Int32	每手数量（沪深为 0，美股为 1，期货为 0）
Cur	Int32	现价
Time	Int32	报价最后更新时间

下面简单讲解一下如何获取这个行情。轮询下单部分和前面讲解的轮询处理一样，不再赘述。行情的获取需要两个步骤，第一步是行情订阅，第二步是行情解读。如果超过订阅数量，可以使用第 1006 号协议注销，或者每次都注销上一个订阅号。对于返回值，直接使用 `json_decode` 函数就可以了，核心代码如下。

```
<?php
.....
//行情
function getFUTU($code){
    $ip = '127.0.0.1';
    $port = 11111;
    $socket = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);
    $result = socket_connect($socket, $ip, $port);
    $in = '{"Protocol":"1001","ReqParam":{"Market":"1","Stock
Code":"' . $code . '"},"Version":"1"}'."\r\n";
    if(!socket_write($socket, $in, strlen($in)) ) {
    } else {
        ob_flush();flush();
    }
    $que = socket_read($socket, 8095, PHP_NORMAL_READ);
    socket_close($socket);
    return $que;
}

//订阅
function fu($code){
```

```

        $ip = '127.0.0.1';
        $port = 11111;
        $socket = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);
        $result = socket_connect($socket, $ip, $port);
        $in =
'{"Protocol":"1005","ReqParam":{"StockSubType":"1","Market":"1",
"StockCode":"" . $code . ""},"Version":"1"}' . "\r\n";
        if(!socket_write($socket, $in, strlen($in)) ) {
        } else {
            ob_flush();flush();
        }
        $que = socket_read($socket, 8095,PHP_NORMAL_READ);
        socket_close($socket);
        sleep(1);
    }

    //取消订阅
    function del($code){
        $ip = '127.0.0.1';
        $port = 11111;
        $socket = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);
        $result = socket_connect($socket, $ip, $port);
        $in = '{"Protocol":"1006","ReqParam":{"StockSubType":"1",
"Market":"1", "StockCode":"" . $code . ""},"Version":"1"}' . "\r\n";
        if(!socket_write($socket, $in, strlen($in)) ) {
        } else {
            ob_flush();flush();
        }
        $que = socket_read($socket, 8095,PHP_NORMAL_READ);
        socket_close($socket);
        sleep(1);
    }

    //返回腾讯股票行情
    fu('00007');
    $json = getFUTU('00007');
    //打印
    print_r(json_decode($json,true));
    ?>

```

5.2 海量图片生成系统

在前面的 3.13 节中，开发了一个图片管理模块组，而在实际项目中，特别是为了满足深度学习的需要而进行的海量图片生成和管理，需要一个更加专业和独立的系统，这个系统可以使用 C#或 Java 开发，也可以使用 PHP 来完成类似的工作，比如开发一个独立的海量图片生成系统。

1. 第三方数据库

首先需要一个专业的第三方数据库，比如这个数据库叫作 data_day，它的表结构是按照股票代码建立的，每只股票对应一张表，比如 k_000002 就表示万科 A 的日 K 线行情，表结构如表 5-6 所示。

表 5-6 A 股日 K 线行情表的表结构

字段	类型	长度	默认	注释	主键
date	varchar	255	0	日期	*
open	float	0	0	开盘价格	*
close	float	0	0	收盘价格	*
high	float	0	0	最高价格	*
low	float	0	0	最低价格	*
volume	float	0	0	成交量	

2. 数据库连接测试

数据库有了表结构后，接下来要测试一下，看看数据库是否能够连接成功，或者是否有权限查询对应的表，测试代码可以很简单，比如如下代码。

```
testmysql.php
<?php
//在下面输入想要连接的数据库地址、账户和密码
$MYSQL = new mysqli('127.0.0.1:3309','readonly','readonly');
$MYSQL->select_db('hist_data');
if($MYSQL->connect_errno){
    echo 'error';
}else{
    echo 'ok';
}
```

```
}
?>
```

3. 任务列表

数据库连接成功后，就可以进行任务轮询了。为了简单起见，仍然可以使用 3.12 节中的 `image_work` 表，方便进行多进程管理，当然这时表中的 `dataid` 和 `username` 等已经没有意义了，不过如果你愿意，还是可以把 `ai_count` 数据库中的 `data_set` 表一起复制到新项目中。由于这是一个新项目，所以最好有一个对应的数据库，就叫它 `ai_images` 好了。

接下来就要添加对应的记录了，在添加记录之前，我们不妨设想一下，一共有 2000 多只股票，分成若干个进程进行批量处理，每个进程至少要处理几百只股票，可是 `image_work` 一次只能处理一张表，怎么办呢？或者说怎样才能够让程序自动查询和更新 `image_work` 记录呢？别着急，一步步来。

首先，返回 `data_day` 数据库中所有的表，由于这些表名就是一只只股票行情表的表名，所以可以通过遍历这个表名来查询所有的表。为了统计方便，可以先把所有的表名全部逐行追加到一个叫作 `mytables.txt` 的文件中，追加代码如下。

```
tables.php
<?php
$MYSQL = new mysqli('127.0.0.1:3309','readonly','readonly');
$MYSQL->select_db("information_schema");
$sql = "SELECT `table_name` FROM `tables` WHERE `table_schema`
= 'data_day'";
$tables = DataList($sql,$MYSQL);
// 全部返回并进行处理
function DataList($s,$mysql){
    $q = $mysql->query($s);
    $data = array();
    if($q){
        while($row = mysqli_fetch_assoc($q)) {
            $data[]=$row;
        }
    }else{
```

```

        $data = 0;
    }
    return $data;
}
$txt = '';
foreach($tables as $v){
    //判断是否为需要的表名，这里可以使用正则 preg_match
    $table_name = $v['table_name'];
    if(preg_match("/^k_\d*$/",$table_name)){
        $txt .= $table_name . "\r\n";//换行符
    }
}
file_put_contents('mytables.txt',$txt);
?>

```

其次，分配任务。由于有 2000 多只股票，每只股票成立的日期有早有晚，所以在查询的时候，可以给出一个区间，比如最近 3 年的股票行情。要查询所有的历史行情，最简单的方法就是凭感觉来分配任务，或者再次统计一下所有表的记录数量，然后分别累加至想要的平均数。这里采用的是区间数据，大体每只股票的记录数都差不多，因此就按照股票的序号顺序分段分配。

现在假设要分 3 个进程来做，也就是说要把 mytables.txt 文件中的数千只股票至少分成 3 部分，比如每个进程负责 100 或 800 只股票，每个进程负责 100 只股票可以多次轮询完成，每个进程负责 800 只股票可以一次轮询完成，如果磁盘空间较小，可以每次轮询的数量少一点。

以每个进程负责 100 只股票为例，把 mytables.txt 文件中的第一个 100、第二个 100、第三个 100 依次复制或截取到 1_tables.txt、2_tables.txt、3_tables.txt 文件中，其中文件前缀分别对应 image_work 表中的 id。

接下来要做的工作就是把每个任务文件中的第一行（也就是第一个表名）当做初始的表，人为地写入 ai_images 数据库的 image_work 表中。这里主要关注 id、k、name、path、sql 和 working 字段。其中，id 是任务 id，要和 1_tables.txt 的 id 前缀一致；name 是搜索时用到的表名；path 是文件存储的位置，一般的存储格式是“任务 id/表名”，如 img/k/1/k_000001；sql

是要对行情表所使用的查询语句，查询的结果要符合前端 K 线需要的数据格式。可以使用一些 SQL 函数，比如这里使用了 `replace` 函数，注意语句 `replace(replace('time',' ','_'),':','') 'date'`，中间空格的用法同 AS，也就是说等同于 `replace(replace('time',' ','_'),':','') AS 'date'`。

'date'、'open'、'close'、'low'和'high'这 5 个字段是前端需要用到的，在这里顺序也很重要，顺序不能错，否则 K 线渲染时就可能出现错误，因此必须和前端的要求相一致。SQL 代码如下。

```
SELECT replace(replace('time',' ','_'),':','') 'date', 'open',  
'close','low','high' FROM 'k_000001' WHERE 'time' > '2013-08-05' AND  
'time' < '2017-05-05'
```

其中，`working` 字段是指工作状态，0 为停止工作，1 为工作中，2 为暂停。

填写好任务记录后的 `image_work` 表如图 5-3 所示。

id 工作记录ID	username 用会员账户	dataid 数据库接口ID	type 图片格式	k K线类型	fun 被调用的函数名称	up 上涨时间跨度	down 下跌时间跨度	unit K线形态周期类型	width 宽度像素	height 高度像素	name 命名规则
1	dong	4	png	black	NULL	1	1	日	1000	600	k_000001
2	dong	4	png	black	NULL	1	1	日	1000	600	k_002610

图 5-3 `image_work` 表

4. 遍历任务

在 `image_work` 表中填写好任务记录后，就可以通过统一的调度页面遍历 `image_work` 表中的任务，也可以通过指定任务 `id` 的方法手动启动任务进程，比如在浏览器地址栏中输入“`http://localhost/images/test.php?id=1`”来启动任务 1，然后分别将 `id=1` 换成 `id=2`、`id=3`，就可以启动任务 2 和任务 3 了。`test.php` 响应代码如下。

```
test.php  
<!doctype xmlns="http://www.w3.org/1999/xhtml">
```



```

<html>
<head>
<meta charset="utf-8">
<title>生成图片</title>
</head>

<body>
<?php
//做一个连接, 测试图片生成
$id = $_GET['id'];
include 'config.php';
$username = 'dong';
if($username && $id){
    //指定任务 id
    $data = DataList("SELECT * FROM `image_work` WHERE `username`
= '$username' AND `working` = 1 AND `id` = $id LIMIT 1", $MYSQL);
    $show = $data[0];
    ?>
    <!--传参-- >
    <a href="k/<?php echo $show['k'] . '_' .
$show['id']; ?>.php?sql=<?php echo $show['sql']; ?>&id=<?php echo
$show['id']; ?>&type=<?php echo $show['type']; ?>&width=<?php
echo $show['width']; ?>&height=<?php echo $show['height']; ?>
&path=<?php echo $show['path']; ?>&maxfile=<?php echo $show
['maxfile']; ?>&slep=<?php echo $show['slep']; ?>&dataid=<?php
echo $show['dataid']; ?>&fun=<?php echo $show['fun']; ?>&up=<?php
echo $show['up']; ?>&down=<?php echo $show['down']; ?>"
id="myhref">开始生成图片</a>

    <?php
    }else{
        echo 'no login';
    }
    ?>

    <script type="text/javascript">
    window.setTimeout(function(){
        document.getElementById("myhref").click();
    },1000);
    </script>

```

```
</body>
</html>
```

在上面的代码中，语句“include 'config.php'”仍然是导入原来的那个配置文件，为了简单起见，可以将之前的拿过来用，这样只要修改'database'=>'ai_images'这个数组，就可以直接连接 ai_images 数据库了，代码如下。

```
config.php
<?php
.....
$data = array (
    'hostname' => 'localhost',
    'database' => 'ai_images',
    'username' => 'root',
    'password' => '',
    'charset'  => 'utf8',
    'type'     => 'mysql',
    'debug'    => true,
    'pconnect' => 0,
    'autoconnect' => 0
);
$MYSQL = new mysqli($data['hostname'],$data['username'],$data
['password']);
$MYSQL->select_db($data['database']);
$MYSQL->set_charset($data['charset']);
.....
?>
```

另外，在 test.php 页面中调用的文件名格式为：“black_1.php? sql=sql 语句&id=任务 id&type=图片类型&width=图片宽度&height=图片高度&path=图片路径&maxfile=最大文件数&slep=休眠时间&dataid=数据接口&fun=函数&up=连续上升数量&down=连续下降数量”。

black_1.php、black_2.php 和 black_3.php 这几个文件实际上就是一个 K 线渲染的页面，在图片生成中已经介绍过，这里主要讲解几个关键节点，代码如下。

```
black_1.php
<?php
```

```
//返回参数, 没有就返回默认值, 客户端提交什么就返回什么
?>
<!doctype html>
```

引用 echarts.js 和 jquery.min.js 文件, 初始化的代码如下。

```
.....
<script type="text/javascript" src="js/echarts.js"></script>
<script type="text/javascript" src="js/jquery.min.js"></script>
.....
</head>

<body>
```

设置图像大小的代码如下。

```
<div id="container" style="width:<?php echo $width; ?>px;
height:<?php echo $height; ?>px;"></div>
<div id="test"></div>
<input type="button" onClick="save_image()" value="保存" />
<script type="text/javascript">
//返回对象 id
var dom = document.getElementById("container");
var myChart = echarts.init(dom);
//以 Base64 方式上传图像
save_image = function(){
    picBase64Info = myChart.getDataURL();
    dom.get;
    //通过 xmlhttp64 位上传
    //document.getElementById("container").innerHTML =
picBase64Info;
    $.post("../post.php", {data:picBase64Info,path:path,maxfile:
maxfile,imgname:imgname,label:label},function(result){
        //成功上传后, 开始上传下一个图像
        if(result){//将数据显示到测试文本框中, 以方便观察
            document.getElementById("test").innerHTML =
result;
            post_sql();
        }
    });
}
```

```

//请求行情记录
post_sql = function(){
    // 开始数据请求
    $.post("../sql.php",{dataid:dataid,sql:sql,workid:workid,
slep:slep,fun:fun,down:down,up:up},function(result){
        //截取最后 3 个字符串,主要是为了确定格式
        //alert(result);
        $len = result.length;
        $result = result.substr(0,$len-4);
        $result = $result + ']';
        label = result.substr($len-3,2);
        //alert($result)用于调试
        if(result != 0){
            //document.getElementById("test").innerHTML =
$result;

            woking(eval($result));
            //每秒生成 10 张图像
            timer = window.setTimeout("save_image()",100);
        }else{
            //如果等于 0,则表示记录为空,搜索完成,停止任务,这时候要换
            下一个表了,换表的代码交给 ajax.php,详见 ajax.php 文件

            $.get("../ajax.php",{n:1,i:workid,sql:sql},function(result){
                if(result == 1){
                    //跳转到 test.php
                    window.location.href = '../test.php?id=' +
workid;

                    }else if(result == 0){
                        alert('停止任务');
                    }else{
                        document.getElementById("test").innerHTML
= result;
                    }
                });
            }
        });
    }
    post_sql();

```

下面给出一个完整的 JavaScript 代码，使读者能够更加清楚、详细地了解使用 echarts.js 绘制 K 线的流程和要素。需要注意 3 个地方，第 1 个是 rawData.map 方法，方法中 return 数组的顺序默认和 SQL 语句返回的顺序一致，如果不一致可以调整；第 2 个需要注意的是“data: ['日 K']”和“name: '日 K'”数组，如果两者不一致，是不会显示日 K 字样的标题的，当然，如果就是不想显示，也是可以的；最后一个需要注意的是 echarts.js 渲染图片一般是通过 option 数组进行设置的，然后通过 setOption(option, true)方法渲染，要想修改样式，修改 option 数组参数就可以了。完整的 JavaScript 代码如下。

```
black_1.php
.....
function woking(rawData){
var app = {};
option = null;
app.title = '';
var dates = rawData.map(function (item) {
    return item[0];
});
imgname = dates[0];
var data = rawData.map(function (item) {
    return [+item[1], +item[2], +item[3], +item[4]];
});
option = {
    backgroundColor: '#21202D',
    legend: {
        data: ['日 K'],
        inactiveColor: '#777',
        textStyle: {
            color: 'fff'
        }
    },
    tooltip: {
        trigger: 'axis',
        axisPointer: {
            animation: false,
            lineStyle: {
                color: '#376df4',
```

```

        width: 2,
        opacity: 1
    }
}
},
xAxis: {
    type: 'category',
    data: dates,
    axisLine: { lineStyle: { color: '#8392A5' } }
},
yAxis: {
    scale: true,
    axisLine: { lineStyle: { color: '#8392A5' } },
    splitLine: { show: false }
},
dataZoom: [{
    textStyle: {
        color: '#8392A5'
    },
    handleIcon: 'M10.7,11.9v-1.3H9.3v1.3c-4.9,0.3-8.8,4.4-8.8,9.4c0,5,3.9,9.1,8.8,9.4v1.3h1.3v-1.3c4.9-0.3,8.8-4.4,8.8-9.4C19.5,16.3,15.6,12.2,10.7,11.9z M13.3,24.4H6.7V23h6.6V24.4zM13.3,19.6H6.7v-1.4h6.6V19.6z',
    handleSize: '80%',
    dataBackground: {
        areaStyle: {
            color: '#8392A5'
        },
        lineStyle: {
            opacity: 0.8,
            color: '#8392A5'
        }
    },
    handleStyle: {
        color: '#fff',
        shadowBlur: 3,
        shadowColor: 'rgba(0, 0, 0, 0.6)',
        shadowOffsetX: 2,
        shadowOffsetY: 2
    }
}]

```

```

    }
  }, {
    type: 'inside'
  }],
  animation: false,
  series: [
    {
      type: 'candlestick',
      name: '日 K',
      data: data,
      itemStyle: {
        normal: {
          color: '#FD1050',
          color0: '#0CF49B',
          borderColor: '#FD1050',
          borderColor0: '#0CF49B'
        }
      }
    }
  ]
};
if (option && typeof option === "object") {
  myChart.setOption(option, true);
}
}
</script>
</body>
</html>

```

下面讲解系统的另一个重点部分，这部分主要集中在 `sql.php` 代码中，之所以重要是因为这部分内容主要是为了满足监督学习进行图片分类，比如把涨幅为零的图片放到文件夹 20 中，把涨幅为 1% 的图片放到文件夹 21 中，把跌幅为 1% 的图片放到文件夹 19 中，以此类推。另外，为了方便进行简单分类，还可以通过传参的方法进行简单的连涨、连跌的图片生成，而不是简单地生成单根日 K 线图。`sql.php` 代码如下。

```

sql.php
<?php
//基本设置和参数初始化

```

```

.....
$label = 20;
$sid = 'sid'.$id;
$lid = 'log'.$id;
$i = 0;

if(!$_SESSION[$sid]){
    $set =
array('ip'=>'192.168.50.13:3309','user'=>'readonly','password'=>'re
adonly', 'database'=>'day_data');
    //连接数据接口
    .....
}else{
    //查询行情记录，并存入 SESSION 数组，减少数据库访问，提高效率，只是
内存要足够大才行
    $_SESSION[$sid] = MyDataList($sql,$CONN);
}
}
//返回数据
function MyDataList($s,$mysql)
{
    $q = $mysql->query($s);
    $data = array();
    if($q){
        while($row = mysqli_fetch_assoc($q)) {
            $data[] = array_values($row);
        }
    }else{
        $data = 0;
    }
    return $data;
}
}
    
```

上述代码默认通过 SESSION 数组返回需要的行情记录，如果 SESSION 数组返回当前记录，则查询并截取记录后，重新保存 SESSION 数组；如果 SESSION 数组不存在，则重新查询行情数据库。另外，还需要注意一下步长程序，也就是希望 K 线图默认展示几天的 K 线，是 20 根、5 根，还是更多。

为了简便起见，单独写了一个叫作 mydata 的函数，函数中有很大大一部

分是根据设置的函数名生成的指定类型的标签（这个标签实际上就是存储位置，方便以后进行有监督学习），代码如下。

```
sql.php
.....

//如果存在，查找日志是否存在
$i = $_SESSION[$lid];
if(!$i){
    $i = 0;
}
$data = mydata($_SESSION[$sid],$slep);
if($data){
    $newarr = array_values($data);
    //自加1
    $_SESSION[$lid]++;
    $newarr[] = $label;
    //增加标签
    echo json_encode($newarr);
}else{
    //清除工作日志
    //print_r($_SESSION[$sid]);
    $i = -1;
    unset($_SESSION[$lid]);
    //清除SESSION数组
    unset($_SESSION[$sid]);
    echo 0;
}

function mydata($session,$slep){
    global $i;
    global $label;
    global $fun;
    global $up;
    global $down;
    // 获得区间
    $limit = $i++;
    $end = $limit + $slep;
    if($i!= -1){
        $ii = 0;
        $data = array();
        //调用函数
```

```

switch($fun){
    case 'up_down':
        //默认方法
        foreach($session as $v){
            //遍历记录
            $ii++;
            //开始计算标签
            if($ii > $limit && $ii<=$end){
                $data[]=$v;
            }
            if($session[$end+$up][2]){
                //截取数组中的指定元素
                $number = ($session[$end+$up][2]-
$session[$end][2])/ $session[$end][2]*100;
                switch($number){
                    case $number >= -10 && $number <-9:
                        $label = 10;
                        break;
                    .....
                    case $number >= 9 && $number <10:
                        $label = 30;
                        break;
                }
            }
        }
        break;
    case 'up_up':
        //上涨函数
        foreach($session as $v){
            //遍历记录
            $ii++;
            //开始计算标签
            if($ii > $limit && $ii<=$end){
                if($session[$end+$up][2] && $session
[$end-$down][2]){
                    //计算收盘价
                    //后Y日涨幅3%以上
                    $back = ($session[$end+$up][2]-
$session[$end][2])/ $session[$end][2]*100;
                    //前N日跌幅10% 以上

```

```

        $front = ($session[$end][2] -
$session[$end - $down][2])/($session[$end - $down][2]*100;
        //截取数组中的指定元素
        //这里需要人为设置
        if($back >= 1 && $front <= -3){
            $label = 30;
        }else{
            $label = 40;
        }
    }
    $data[]=$v;
}
}
break;
default:
    //没有标签,或者标签为20
    foreach($session as $v){
        //遍历记录
        $ii++;
        //开始计算
        if($ii > $limit && $ii<=$end){
            $data[]=$v;
        }
    }
}
}else{
    $data = 0;
}
return $data;
}
?>

```

post.php 代码和图片生成的代码相似,只不过多了一个标签的处理方法,也就是说,同样的目录中可能多了一些类似“20”“30”“40”这样的子文件夹。获取这些子文件夹的主要代码如下。

```

post.php
<?php
.....
$label = $_POST['label'];

```

```
$path = $_POST['path'] . '/' . $label;
.....
?>
```

当进程循环读取完一只股票的行情记录时，也就是记录为零时，需要切换到下一只股票，这时就可以用 Ajax 方法请求一个名为 ajax.php 的文件，它的工作非常简单，就是通过任务 id 从指定的任务文件 _tables.txt 中返回第一行内容，然后用这行内容作为表名更新对应任务 id 的字段，在做更新时需要注意一些特殊字符的处理，很可能这些特殊字符来自于你的 SQL 代码。完整代码如下。

```
ajax.php
<?php
include 'config.php';
//更新生成图片的工作状态
$i = $_GET['i'];
$sql = $_GET['sql'];
$username= 'dong';
$MYSQL->select_db("ai_tree");
//返回第一行并删除
$txt = 'k/' . $i . '_tables.txt';
$tables = file($txt);
$tablename = array_shift($tables);
$tablename = str_replace(array("\r\n", "\r", "\n"), "",
$tablename);
//使用分隔符连接并保存
$newtxt = implode('',$tables);
$f = file_put_contents($txt,$newtxt);
//替换表名
$sql = preg_replace('/(FROM)\s+(\`)\w+(\`)/i','FROM ' .
"``$tablename`",$sql);
//判断是否需要转义
$sql = str_replace("'",`',$sql);
$path = 'img/k/' . $i . '/' . $tablename;
if($tablename){
//这里就是更新，和前面手动录入的任务记录几乎是一样的
if($MYSQL->query("UPDATE `image_work` SET `sql` = '$sql', `name`
= '$tablename', `path` = '$path', `working` = '1' WHERE `id` = $i AND
`username` = '$username' LIMIT 1")){
```

```

    echo '1';
}else{
    echo $MYSQL->error;
};
}else{
    echo '0';//表示完成
}
?>

```

在本节的最后，通过一个开发流程图作为小结，海量图片生成系统开发流程图如图 5-4 所示。

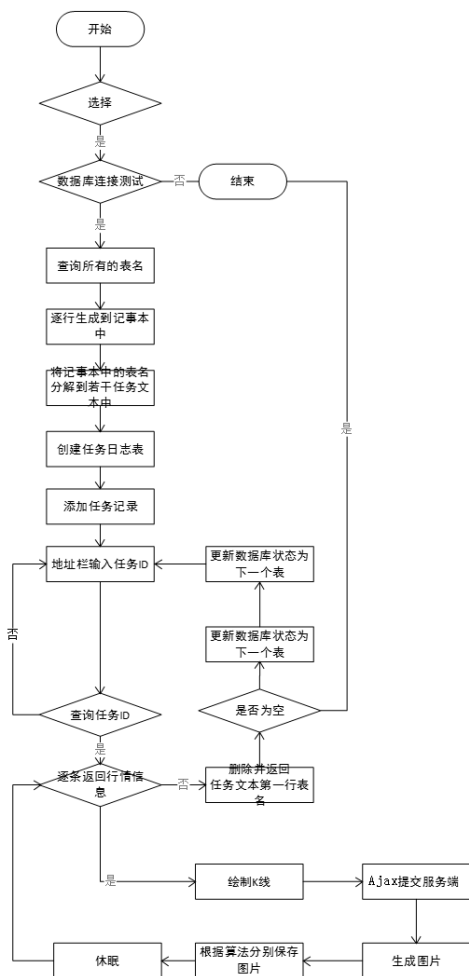


图 5-4 海量图片生成系统开发流程图

5.3 机器学习系统

人类的大部分经验都是通过历史数据得来的，特别是那些需要后天培养的技能，比如聊天和工作，尤其是工作。我们都知道，一般在招聘程序员时，有 3~5 年工作经验的程序员薪资就比有 1~3 年工作经验的程序员薪资高不少，而这部分所谓增加的薪资就是工作经验的价值，有些时候招聘需求上明确地写明拥有某项工作经验者优先。

这种经验的获得最主要的方法就是学习，同样的工作用心工作的人往往比工作敷衍的人更容易获得经验。

从语义上来讲，经验就是那些我们经历并验证过的事物。从科学上来讲，我们的经验更偏向于统计科学。

统计无疑是一门非常严谨的科学，但是统计学上仍然有一些小问题需要完善，而正是这种小问题才是我们日后工作中需要注意的地方。

统计的第一个问题也是最重要的问题就是样本选择，很多专家做出来的看似很漂亮的统计图之所以让普通大众感觉不真实，问题往往就出现在样本选择的环节上。下面举两个小例子。

比如，某些知名院校或培训机构，经常通过学生就业率来衡量教学成果，统计的结果就是“参加就业培训的人的就业率比不参加就业培训的人的就业率明显偏高”，为什么呢？真的就是培训的效果非常好吗？实际上不完全是，因为参加培训的人本来就是想就业的，如果 he 不想就业，就不会浪费时间来培训了。这就是样本选择的问题，除非把所有的人都召集起来，然后进行随机分配，一半人参加培训，另一半人不参加培训，当然这是不太可能的。

再比如，美国知名杂志评选出“最安全的城市”和“最不安全的城市”，竟然都是一些小城市，为什么呢？真的是这个小城市很安全（或者很不安全）吗？实际上不是这样的，就拿不安全城市来说，在小城市里只要发生一起暴利事件，就可能成为一个较大的分子，因为它的分母也就是小城市

的人口数（小城市的人口数有几十万）很小；反观大城市，由于人口基数大（大城市的人口有数百万甚至更多），即使发生一两起暴利事件，其比值也很小。

那么问题出来了，程序员或交易员应该怎么办呢？虽然科学家暂时还没有找到完美的方法，但是我们却可以改变统计思路。比如，建议连续多年（20年以上）统计一个城市的数据，如果这个城市历史表现良好，那么就比较有说服力了。同时，还要注意波动率，比如同样是股票的波动率，我国由于货币贬值的原因，10年前的购买力和20年前的购买力是不同的，如果只看波动率，那么就会发现时间越久远，波动率越小，难道因此断定20年前的股票行情不疯狂吗？不能。那怎么办？我们是不是应该加入一些标准差和基准年限呢？

统计的第二个问题就是模型的选择，很多做统计的人经常会问的一个问题就是应该用哪种模型，哪种算法更适合现在这样的数据，是方差还是均差，是中数还是众数。

当然，上述两点都是题外话，也是众多大数据公司竞争的法宝，无所谓好坏，就像我国的武术一样，五花八门各有所长，没有什么是一成不变的，具体就看我们怎么运用了。作为一名系统开发人员，只要知道数据是基础、算法是灵魂，不同的数据和算法会得到不同的结果就可以了。

既然算法是灵魂，那么就先介绍一下算法，只有明白了算法的原理，才能更好地进行源码设计。在统计学中，有一个非常有名的贝叶斯定理。贝叶斯定理是在18世纪由英国牧师贝叶斯先生首先提出来的，但是真正让它发扬光大的却是拉普拉斯先生。拉普拉斯先生以贝叶斯命名这个算法并推广开，就好像基督教实际上是由圣保罗（《新约》最主要的作者）开创的一样。所以，我们现在学习的贝叶斯定理实际上是由拉普拉斯先生编著后的内容。贝叶斯定理非常有意思，下面先讲一个小案例。

张医生开了一家诊所，他发现上个月有100名患者，其中15名患者为流行性感冒，20名患者有发烧症状，10名患者既感冒又发烧。因此，他推断感冒中发烧病人的概率为 $10/15 = 2/3$ 。同样，我们还能推断出以下结果。

(1) 既感冒又发烧 (10/100) = 感冒 (15/100) × 感冒中发烧 (10/15)

(2) 既感冒又发烧 (10/100) = 发烧 (20/100) × 发烧中感冒 (10/20)

(3) 发烧 (20/100) × 发烧中感冒 (10/20) = 感冒 (15/100) × 感冒中发烧 (10/15)

(4) 等式两边同时除以发烧的概率 (20/100):

发烧中感冒 (10/20) = 感冒 (15/100) × 感冒中发烧 (10/15) / 发烧 (20/100)

(5) P 表示概率, A 表示感冒, B 表示发烧, 于是可以得到下面这个公式:

$$P(A|B) = P(A) \times P(B|A) / P(B)$$

上面这个公式就是经典的贝叶斯定理。通过贝叶斯定理, 我们可以做很多事情, 比如在股票预测中, 就可以通过贝叶斯定理来预测昨日涨停与次日继续上涨的概率。也许你会问, 干嘛不直接统计一下, 为什么还要用这么复杂的公式呢?

好吧, 好像很多交易策略都是这样直接统计出结果的, 那么就先按照这个想法进行一下传统的统计。

先要有一个符合要求的数据, 比如一张有关股票涨停的 CSV 表, 如表 5-7 所示, 字段对应的格式分别为 code (股票代码)、date (涨停日期)、close (收盘价)、high (后几日最高价) 和 hclose (后几日最高收盘价)。

表 5-7 股票涨停的 CSV 表 (data/k.csv)

code	date	close	high	hclose
k_000002	2006/5/8	2.282	2.388	2.308
k_000002	2006/11/30	4.092	4.56	4.338

k.csv 表中给出了后几日最高价和后几日最高的收盘价, 从经验上来讲, 后几日最高价一定是上涨的概率大, 而收盘价就要比最高价少一些了, 当然就像我们前面说的, 选择数据是一门艺术, 具体可以根据实际情况进行基本的数据加工, 这里选取的主要是 A 股 2006 年以来的 3000 多只股票涨

幅超过 9.7% 的前复权数据。

下面统计一下，看看涨停之后的股票在一周内继续上涨的概率是多少，主要代码如下。

```
count.php
<?php
//CSV 数据文件
$csv = 'data/k.csv';
$arr = file($csv);
//输出行数
$count = count($arr);
//echo $count;
//初始化数组
$n = array(0,0,0,0);
foreach($arr as $v){
    $r = explode(',',$v);
    //给出上涨区间
    //给出一个比率
    //echo $r[3] . '-' . $r[2];
    $number = ($r[3]-$r[2])/$r[2]*100;
    switch($number){
        //涨幅 0~10%、-10%~0、大于 10%、小于-10%四个区间的概率
        //这里也可以使用众数
        case $number <=-10:
            $n[0]++;
            break;
        case $number <= 0 && $number >-10:
            $n[1]++;
            break;
        case $number >0 && $number < 10:
            $n[2]++;
            break;
        case $number >= 10:
            $n[3]++;
            break;
    }
}
//输出统计结果
```

```
//print_r($n);
echo '在' . $count . '只涨停的股票中, 后五日最高价变化情况如下: <br>';
echo '下跌大于 10%的约: ' . round($n[0]/$count*100,3) . '%<br>';
echo '下跌介于 0~10%的约: ' . round($n[1]/$count*100,3) . '%<br>';
echo '上涨介于 0~10%的约: ' . round($n[2]/$count*100,3) . '%<br>';
echo '上涨大于 10%的约: ' . round($n[3]/$count*100,3) . '%';
?>
```

结果很有意思, 果然不出所料, 股票涨停之后的未来 5 日内还有极大的可能性涨幅高达 10%。令人比较意外的是, 最高收盘价居然也有接近 80% 的可能性涨幅超过 10%, 如表 5-8 所示。

表 5-8 统计结果表

涨跌百分比	最高价	最高收盘价
下跌大于 10%的	0.517%	2.591%
下跌介于 0-10%的	0.378%	2.544%
上涨介于 0-10%的	14.557%	16.118%
上涨大于 10%的	84.548%	78.747%

通过表中的数据, 我们不难看出为什么涨停之后, 次日仍然还有很多人愿意以差不多的价格买入股票的原因了。

虽然人们经常拿统计比率来代替概率, 但是概率并不等于比率。比率是已知的且已经发生的, 而概率是未知的且可能发生的。用已知求出未知就是交易员主要思考的事情, 尤其要考虑当已知比率发生改变时, 未知概率将会如何变化。贝叶斯定理是一个非常基础的定理, 通过张医生的例子, 我们成功地推导出了发烧中感冒的概率公式, 既发烧中感冒的概率 (10/20) = 感冒概率 (15/100) × 感冒中发烧的概率 (10/15) / 发烧的概率 (20/100)。

$$15/100 \times 10/15 \div 20/100$$

$$= 3/20 \times 2/3 \div 1/5$$

$$= 1/10 \div 1/5$$

$$= 1/2$$

那么就利用这个公式做一个有趣的计算。我们都知道医生在诊断重大疾病 (比如癌症) 时, 都是需要反复确诊的, 甚至患者家属都感觉有点过

度诊断了。那么他们为什么还要这么做呢？答案就在贝叶斯定理中。

已知每 100 个人中就有 1 个癌症患者，也就是说一个人患癌症的可能性为 1/100，医生诊断的准确率为 99%，误诊率为 1%，也就是说：

- 被诊断为阳性（这里特指患病的人）的人中的 99% 的人可能是癌症患者；
- 同理可以知道，被诊断为阳性的人中有 1% 的人可能是健康的人；
- 被诊断为阴性（这里特指健康的人）的人中的 99% 的人可能是健康的人；
- 同理可以知道，被诊断为阴性的人中有 1%（即 1-99%）的人可能是癌症患者；
- 癌症患者被诊断为阳性的概率也是 99%。

那么一个普通人被诊断为阳性，他可能是癌症患者的概率是多少？也许你的回答是 99%，但是通过贝叶斯公式，发现结果居然是 50%。

贝叶斯公式：

$$P(A|B) = P(A) \times P(B|A) \div P(B)$$

将该例子带入贝叶斯公式：

$$P(\text{癌症}|\text{阳性}) = P(\text{癌症}) \times P(\text{阳性}|\text{癌症}) \div P(\text{阳性})$$

普通人患癌症的概率为：

$$P(\text{癌症}) = 1 \div 100 = 0.01$$

癌症患者被诊断为阳性的概率为：

$$P(\text{阳性}|\text{癌症}) = 99 \div 100 = 0.99$$

最关键的一步是，普通人诊断为阳性的概率是多少，他的概率应该包括真阳性和假阳性。真阳性等于癌症患者的概率乘以诊断准确率即 0.01×0.99 ，假阳性等于非癌症患者的概率乘以误诊率即 0.50×0.01 。

将两者相加：

$$P(\text{阳性}) = 0.01 \times 0.99 + 0.99 \times 0.01 = 0.0198$$

带入公式：

$$0.01 \times 0.99 / 0.0198 = 0.0099 / 0.0198 = 1 \div 2 = 0.5 = 50\%$$

$$0.50 \times 0.99 = 0.495$$

$$0.495 \div (0.495 + 0.005) = 0.495 \div 0.50 = 99\%$$

但是再带入一次就不一样了，复检的时候，多复查几次，诊断概率居然高于正常的 99%，非常接近 100%。

看来复检还是有道理的。反复检查，效果更真实，可以提高检查准确率。

人们还经常这样写贝叶斯公式：

$$P(B_i | A) = \frac{P(B_i)P(A | B_i)}{\sum_{j=1}^n P(B_j)P(A | B_j)}$$

同样还是张医生的例子，张医生这次又通过病历表查看了一下患者的职业，病历表如表 5-9 所示。

表 5-9 病历表

病号	症状	感冒	职业
1	1	1	农民
2	0	0	工人
.....
100	1	0	工人
100 人（小计）	20 人	15 人	工人、农民

最终统计的结果是，在 100 名患者中，其中 15 名患者为流行性感冒，20 名患者有发烧症状，10 名患者既感冒又发烧，其中有 10 位农民患感冒，有 5 位工人患感冒。

现在又来了第 101 个病人，是一个发烧的农民，请问他患上感冒的概率有多大？

根据贝叶斯定理可得：

$$P(\text{感冒} | \text{发烧、农民}) = P(\text{感冒}) \times P(\text{发烧、农民} | \text{感冒}) \div P(\text{发烧、农民})$$

现在假定“发烧”和“农民”这两个特征是彼此独立发生的，也就是说：

$$P(\text{发烧、农民}|\text{感冒}) = P(\text{发烧}|\text{感冒}) \times P(\text{农民}|\text{感冒})$$

$$P(\text{发烧、农民}) = P(\text{发烧}) \times P(\text{农民})$$

因此，上面的等式就变成了：

$$P(\text{感冒}|\text{发烧} \times \text{农民}) = P(\text{感冒}) \times P(\text{发烧}|\text{感冒}) \times P(\text{农民}|\text{感冒}) \\ \div P(\text{发烧}) \times P(\text{农民})$$

这个等式是可以计算的：

$$P(\text{感冒}|\text{发烧} \times \text{农民}) = 0.15 \times 0.66 \times 0.66 \div 0.2 \times 0.5 = 0.6534$$

因此，这位发烧的农民约有 65% 的可能得了感冒。这就是贝叶斯分类器的基本方法，而我们天真地假设那些特征都是独立发生的方法就称为朴素贝叶斯方法。虽然这个假设有些问题，在现实中不太可能成立，但是这个分类方法非常好用，所以朴素贝叶斯就开始流行起来了，英文 *naive* 也有天真幼稚的意思，也就是说我们天真地认为这些特征是彼此独立的，只有彼此独立才可以大大简化计算过程。

在统计资料的基础上，依据某些特征，计算各个类别的概率，从而实现分类，这种方法就是朴素贝叶斯分类。假设某个体有 n 项特征 (Feature)，分别为 F_1 、 $F_2 \dots F_n$ 。现有 m 个类别 (Category)，分别为 C_1 、 $C_2 \dots C_m$ 。贝叶斯分类器就是计算出概率最大的那个分类，也就是求下面这个算式的最大值：

$$P(C|F_1, F_2 \dots F_n) \\ = P(C) \times P(F_1 F_2 \dots F_n | C) \div P(F_1 F_2 \dots F_n)$$

由于 $P(F_1 F_2 \dots F_n)$ 对于所有的类别都是相同的，所以可以省略，问题就变成了求 $P(C) \times P(F_1 F_2 \dots F_n | C)$ 的最大值。

朴素贝叶斯分类器则更进一步，假设所有特征都彼此独立，因此就变成了求下面的算式：

$$P(F_1 F_2 \dots F_n | C) P(C) = P(F_1 | C) \times P(F_2 | C) \dots P(F_n | C) \times P(C)$$

上式等号右边的每一项都可以从统计资料中得到，由此就可以计算出每个类别对应的概率，从而找出最大概率的那个类。

下面通过一个股票的例子介绍如何使用朴素贝叶斯分类器。股票的样本数据只有涨和跌两种，一种是涨停，另一种是跌停。我们做一个最简单的特征，就是根据昨日涨停的 K 线来判断未来 5 日最高价的涨跌概率。这个做法明显不明智，比较明智的做法是加上公告、财报及均线。但是这样操作起来比较复杂，为了讲解方便，这里仍然采用 k.csv 表做样本数据，只不过每次都随机选择 80% 的数据作为训练（学习样本）用，另外 20% 的数据作为测试（测试数据）用，也就是所谓的二八法则。

根据 k.csv 表数据统计，其中约 10 万次涨停的股票中有 99% 的股票在接下来的 5 天之内最高价是超过涨停价的，假设这类股票为“上涨”；1% 的股票在接下来的 5 天之内最高价是低于涨停价的，假设这类股票为“下跌”。因此可得：

上涨的概率 = 0.99

下跌的概率 = 0.01

接下来，就要判一只涨停的股票未来 5 天之内的最高价是否超过了涨停价。假设某一只股票有以下 3 个特征。

F_1 : 换手率（成交量/流通总股数）。

F_2 : 市净率的倒数（每股净资产/每股的价格）。

F_3 : 是否是龙头股（龙头股票为 1，非龙头股票为 0）。

已知：

$F_1 = 0.01$

$F_2 = 0.4$

$F_3 = 1$

那么该只股票涨停之后，未来 5 天之内的最高价是否能超过涨停价呢？下面就用朴素贝叶斯分类器计算一下，算式如下：

$$P(F_1|C)P(F_2|C)P(F_3|C)P(C)$$

虽然上面这些值可以从统计资料中得到，但是这里有一个问题：F1 和 F2 是连续变量，不适合按照某个特定值计算概率。于是我们根据经验将 F1 分解成 0~0.05、0.05~0.1 和大于 0.1 三个区间，一般一只股票超过 1% 的换手率就已经是很大的交易量了，所以根据经验放大 10 倍，也就是说 1% 的换手率在这里变成 10%。然后计算每个区间的概率，这里既可以凭借经验计算，也可以用条件概率及拉普拉斯算式等进行计算。

在这个例子中，F1 等于 0.1，落在第二个区间，所以在计算时，就使用第二个区间的发生概率。根据统计资料（或者凭借主观经验，认为正相关的就增加一些概率）可得：

$$P(F_1|\text{上涨}) = 0.6, P(F_1|\text{下跌}) = 0.2$$

$$P(F_2|\text{上涨}) = 0.7, P(F_2|\text{下跌}) = 0.3$$

$$P(F_3|\text{上涨}) = 0.6, P(F_3|\text{下跌}) = 0.4$$

这个统计的意思是，在上涨的股票中，60%与换手率有关，70%与市净率倒数有关，60%与是否是龙头股票有关；在下跌的股票中，20%与换手率有关，30%与市净率倒数有关，40%与是否是龙头股票有关。因此：

$$P(F_1|\text{上涨}) P(F_2|\text{上涨}) P(F_3|\text{上涨}) P(\text{上涨})$$

$$= 0.6 \times 0.7 \times 0.6 \times 0.99$$

$$= 0.24948$$

$$P(F_1|\text{下跌}) P(F_2|\text{下跌}) P(F_3|\text{下跌}) P(\text{下跌})$$

$$= 0.2 \times 0.3 \times 0.4 \times 0.01$$

$$= 0.00024$$

可以看到，这只股票未来上涨超过涨停价的概率是下跌低于涨停价概率的 1000 多倍，也就是说换手率在 1%、市净率倒数为 0.4（2.5 的市净率）的龙头股票在涨停之后的未来 5 日之内极有可能最高价超过原涨停价。

当然上面更多的是一个例子，而我们也只是简单地分成了上涨和下跌，实际上也可以把涨幅这种相对连续的数据进行离散化，分成几个阶段。朴

素贝叶斯分类的开发流程如图 5-5 所示。

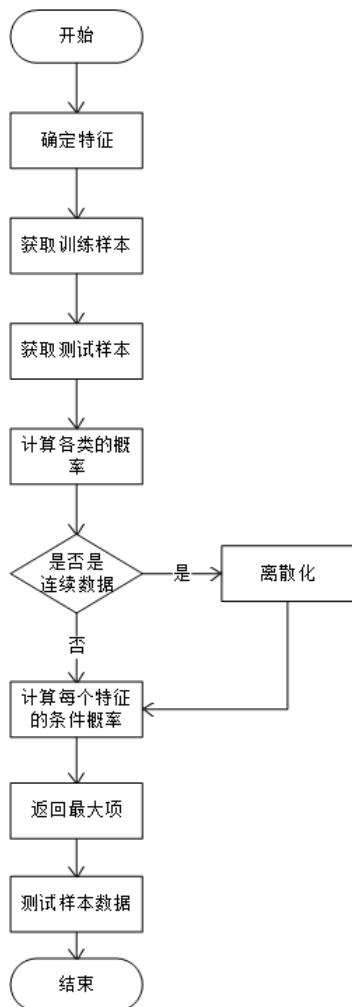


图 5-5 朴素贝叶斯分类的开发流程

除朴素贝叶斯外，还有很多机器学习的算法，比如逻辑回归、反向传播、决策树、蒙特卡洛搜索、神经网络、支持向量机、随机聚类 and 遗传编程等。感兴趣的读者可以自己去看具体的算法原理，只有懂得了原理才能做好机器学习框架。这里给出一张机器学习系统功能的结构流程图供读者参考，方便读者开发自己的机器学习框架，如图 5-6 所示。

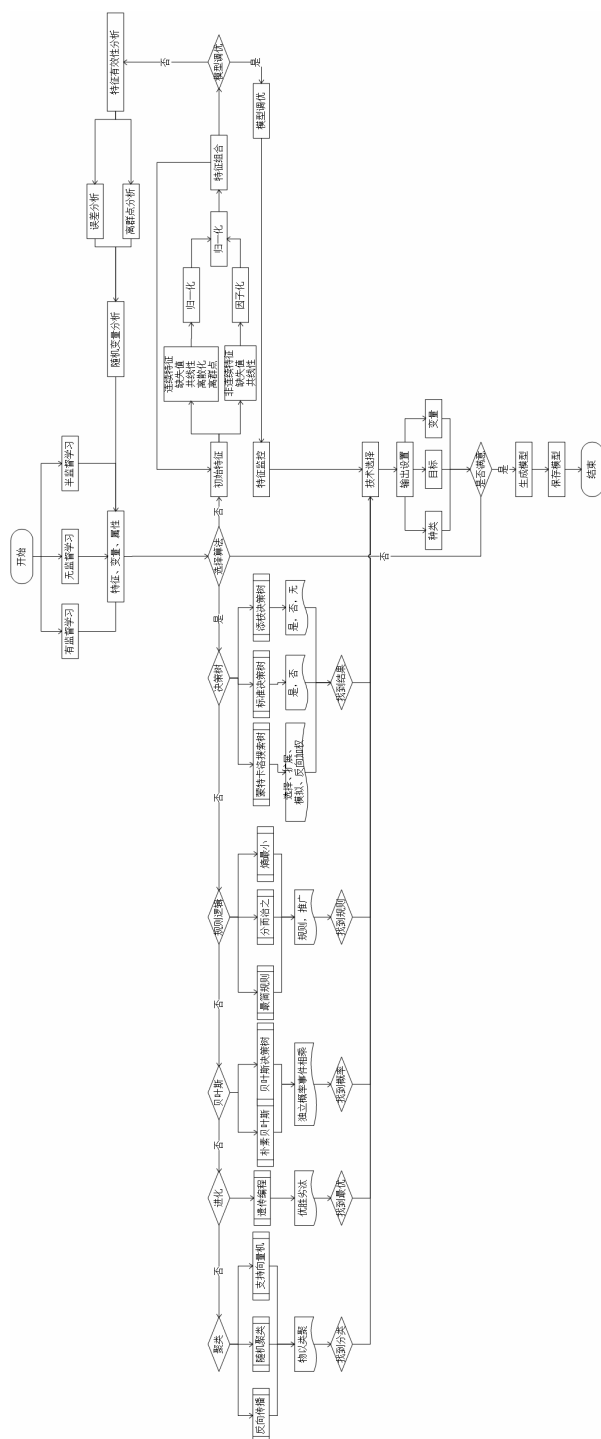


图 5-6 机器学习系统功能结构流程图



附录 A

实际问题

由于专家系统具有不确定性，所以成功的专家系统往往需要利用大量的实际案例进行反复测试，从而发现错误的知识并加以剔除。比如，在实际工作中我们就发现了以下几个有趣的问题。

(1) 每个产品的交易佣金、服务费等会经常变化，尤其是期货，平仓和平今就有很大的变化，产品设计时根本没有想到这一点，所以增加了人为修改具体费用的额外工作。

(2) 模拟盘的部分，无法实时跟进排队中的具体交易，比如同一价位排队的人很多，我们虽然也排队了，但是不知道是否能够交易，只有超过这个价位时才能买进，与实际排队情况不符。

(3) 行情数据有延迟，无法实现高频交易，产品设计时认为专业的数据服务商提供的数据一定是最快最好的，实际运用时发现它们是有差别的，尤其是合并成 K 线之后，分钟线的差异很大。

(4) 在实际运行中，很多基本面和价值投资的专家需要对财报、工商数据等进行专业的分析，目前系统无法满足这些要求。

(5) 产品的修改（包含增加新功能）比想象中的要多，甚至希望第二期的项目能够早日上线。



附录 B

汉语言

每个行业都有自己的专属语言，这些语言在其专业领域往往代表着特殊的意思。作为交易员，除这些行业语言外，经常说的就是自然语言了。为了能够方便地把自然语言和计算机编程语言结合起来，我们需要一个专门的翻译器。

传统最好的做法就是使用字典，字典采用映射的方法，把不同的语句结构映射成不同的计算机语言，比如下面的例子。

自然语言：如果明天下雨，我就买雨伞。

计算机语言：如果（下雨==真）那么{我买雨伞}

把所有的自然语言作为一个字典是很难的，也不现实，所以很多人用到了词库和语法结构，目前看来这个还是很合理的。

虽然有些科学家用深度学习来转换效果更好，但是由于金融行业的这类数据相对匮乏，所以还是采用“语法结构+词库”的方式比较现实。

除此之外，还有一个更加有效的办法就是创造一种新的金融投资行业语言，这种语言语法严谨，内容量化，而又不失灵活。最重要的是学习简单并能够直接翻译成计算机语言。

比如，如果财务数据中的年营业额大于 100 亿元，则在 10 元每股的时候买入 100 手。

这是一个系统工程，是人机交互的一个中间语言。如果是语音，还要注意停顿和语气。下面笔者谈一下自己的一些想法。

(1) 语素是有限的，通过语素之间的组合，词汇可以是无限的。

(2) 语法也是有限的，否则学习起来太难。比如，“的”是包含、“和”是并且等。具体可以参考中文的语法结构。

- 主语：陈述的对象，什么人，什么事物，如提高我的学习能力。
- 谓语：是什么，怎么样，如我今年 30 岁。
- 宾语：回答谁或者什么，如失败是成功的老师。
- 定语：主语的修饰词，如翠绿的柳树。
- 状语：修饰动词和形容词，如他已经上车了。
- 补语：动词、形容词达到的结果，如跑得汗流浹背。

小口诀：

主谓宾定状补，

定语在主宾前，

状语在谓语前，

补语在谓语后。

实词：

- 名词（人物、时间、位置）。
- 动词（变化运动）。
- 形容词（形状、颜色、状态等）。
- 数词（数字）。
- 量词（数量）。
- 代词（人称代词）。

虚词：

- 副词（动词或形容词前面）。
- 介词（在名词或代词前面）。

- 连词（连接作用）。
- 助词（语气词，辅助作用的词）。
- 叹词（表示感叹）。
- 拟声词（模拟事物的声音）。

（3）基于对象，弱类型，参见 JavaScript 的语法结构。

（4）创建一个语法学习机，输入字符串“文法”并输出（正确或错误）。

（5）创建金融行业词库。



附录 C

源码

本书提供可供参考的源代码，源码中剔除了实际项目中的敏感信息，用示例信息代替，比如用户名、密码、策略和对应的数据接口等。

关于邮箱、模拟交易的账户名密码，需要自己申请和修改，源码中的此类账户密码已经修改过，所以不可用。

将 SQL 语句分别导入到对应的数据库，然后配置 host 文件为：

```
127.0.0.1      www.ai.com
```

接着在 Web 服务器（比如 Wampserver64、Apache）中增加一个 `www.ai.com` 的域名，然后将源码复制到对应的 `E:/wamp/www/www.ai.com` 文件夹中，代码如下。

```
<VirtualHost *:80>
    ServerName www.ai.com
    DocumentRoot E:/wamp/www/www.ai.com
    <Directory "E:/wamp/www/www.ai.com">
        #禁止目录浏览
        Options -Indexes +Includes +FollowSymLinks +MultiViews
        #启用.htaccess
        AllowOverride All
        #允许外网访问
        Require all granted
    </Directory>
```

```
</VirtualHost>
```

重启 Wampserver64 后在浏览器的地址栏中输入 `http://www.ai.com/`即可，如图 C-1 所示。

另外，项目拓展 1 为模拟交易系统，源码目录为 `futu`；项目拓展 2 为海量图片生成系统，源码目录为 `images`；项目拓展 3 为机器学习系统，源码目录为 `phpml`。

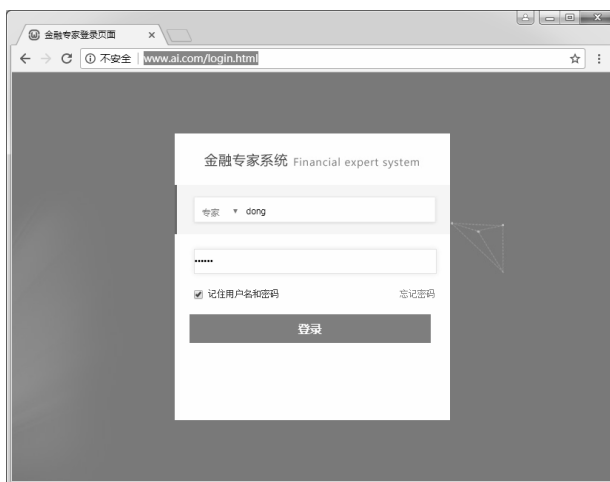


图 C-1 量化投资专家系统登录界面

